AMCC STORAGE

3ware® SAS/SATA RAID Controller

CLI Guide

Command Line Interface

Version 9.5—Supports the 9000 Series (9690SA, 9650SE, 9590SE, 9550SX, and 9500S)

PN: 720-0176-00
September 2007

Copyright

Trademarks

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

About this CLI Guide

3ware SAS/SATA RAID Controller CLI Guide provides instructions for configuring and maintaining your 3ware controller using 3ware's command line interface (CLI).

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

AMCC makes a number of 3ware controller models, including 9000-series (9690SA, 9650SE, 9590SE, 9550SX, and 9500S), and earlier 7000/8000 series boards. The Command Syntax described in this document supports all of those boards, although not all commands are supported on every controller. When a command is not supported for all controllers, the specific model numbers that it applies to are indicated with the command.

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 the following applications:

- 3ware BIOS Manager
- 3DM®2 (3ware Disk Manager)

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

1

Introduction to the 3ware Command Line Interface

The 3ware SATA RAID Controller Command Line Interface (CLI) manages multiple 7000, 8000, and 9000-series 3ware ATA and Serial ATA RAID controllers via a command line or script.

Note: Some CLI commands are supported only for particular models of 3ware RAID controllers. Wherever possible, commands are labeled to indicate when they are supported for only a subset of controllers. For example, commands that apply only to 3ware 9000 series controllers are labeled as such and are not supported for 3ware 7000/8000 controllers. Within the 9000 series, some commands apply to only to models 9690SA, 9650SE, 9590SE, and 9550SX, but not to 9500S, and are so labeled. A few commands apply only to models 9500S, and are labeled as such.

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/support for the latest versions and upgrade instructions.

This chapter includes the following sections:

- "Features of the CLI" on page 2
- "Installing the 3ware CLI" on page 4
- "Working with 3ware CLI" on page 6
- "Understanding RAID Levels and Concepts" on page 9

Features of the CLI

3ware CLI is a command line interface for managing 3ware RAID Controllers. It provides controller, logical unit, drive, enclosure, and BBU (Battery Backup Unit) 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 20.

Supported Operating Systems

The 9.5 version of the 3ware CLI is supported under the following operating systems:

- **Windows®.** Windows XP and Windows Server 2003, both 32-bit and 64-bit.
- **Linux®**. Redhat, SuSE, both 32-bit and 64-bit.

For specific versions of Linux that are supported for the 3ware CLI, see the Release Notes.

Terminology

3ware SAS/SATA RAID Controller CLI Guide 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 and RAID 50 are examples of three-tier units and as such will have sub-sub-units.

Port. 3ware controller models up to the 9650SE series have one or many ports (typically 4, 8, 12, 16, or 24). Each port can be attached to a single disk drive. On a controller such as the 9650SE with a multilane serial port connector, one connector supports four ports. On 9690SA series controllers, connections are made with phys and vports (virtual port).

Phy. Phys are transceivers that transmit and receive the serial data stream that flows between the controller and the drives. 3ware 9690SA controllers have 8 phys. These “controller phys” are associated with virtual ports (vports) by 3ware software to establish up to 128 potential connections with the SAS or SATA hard drives. Each controller phy can be connected directly to a single drive, or can be connected through an expander to additional drives.

VPort. Connections from 3ware 9690SA controllers to SAS or SATA drives are referred to as *virtual ports*, or VPorts. A VPort indicates the ID of a drive, whether it is directly connected to the controller, or cascaded through one or more expanders. The VPort ID allows a drive to be consistently identified, used in a RAID unit, and managed.

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

Warning!

If you have a 7000/8000 controller and are using 3DM, as opposed to 3DM2, AMCC does not recommend installing both 3DM and CLI on the same system. Conflicts may occur. For example, if both are installed, alarms will be captured only by 3DM. You should use either CLI or 3DM to manage your 3ware RAID controllers.

This is not an issue for 3DM2, which works with the 9000-series controllers. It can be installed with CLI.

Installing the 3ware CLI on Windows

3ware CLI can be installed or run directly from the 3ware software CD, or the latest version can be downloaded from the 3ware web site, http://www.3ware.com. Online manual pages are also available in nroff and html formats. These are located in `/packages/cli/tw_cli.8.html` or `tw_cli.8.nroff.`

To install 3ware CLI on Windows

Do one of the following:

- Start the 3ware CD and at the 3ware menu, click **Install Software**.

  Step through the pages of the installation wizard and make sure that Command Line Interface (tw_cli) is selected.

- Copy the file `tw_cli.exe` to the directory from which you want to run the program.

  CLI is located on the 3ware CD in the directory `\packages\cli\windows`

Note: CLI comes in both 32-bit and 64-bit versions. If you are copying the file directly, be sure to copy the correct version for the version of the operating system you are using.

Permissions Required to Run CLI

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

- Administrator
- User with administrator rights
- Domain administrator
- Domain user with Domain Admin or Administrator membership

Without the correct privileges, CLI will prompt and then exit when the application is executed.

If you are uncertain whether you have the correct permissions, contact your network administrator.

To start CLI, do one of the following:

- Start the 3ware CD and at the 3ware menu, click **Run CLI.**
- Or, open a console window, change to the directory where tw_cli is located, and at the command prompt, enter
  `tw_cli`
- OR, double-click the CLI icon in a folder.

  The CLI prompt is displayed in a DOS console window.

Installing the 3ware CLI on Linux

3ware CLI can be installed or run directly from the 3ware software CD, or the latest version can be downloaded from the 3ware web site, http://www.3ware.com.

To install 3ware CLI on Linux

Do one of the following:

- If you are using a graphical user interface such as X Windows, insert and mount the 3ware CD. At the 3ware menu, click **Install Software**. If the 3ware menu does not appear automatically, open a command window and type ./autorun from the CD directory.

  Step through the pages of the installation wizard. On the 3ware Disk Management Tools screen, make sure that Command Line Interface (tw_cli) is selected.

  More detailed information about using this installer is supplied in Appendix B of *3ware Serial ATA RAID Controller User Guide*.

- If no GUI is installed, navigate to packages/installer/linux/x86 (or /86-64, depending on your system) on the 3ware CD.

  Type:

  `./setupLinux_x86.bin -console` or `./setupLinux_x64.bin -console` (depending upon which directory you are in) and press Enter to begin installing the CLI.

  More detailed information about using this installer is supplied in Appendix B of *3ware Serial ATA RAID Controller User Guide*.

- Or, copy the file tw_cli to the directory from which you want to run the program.

  CLI is located on the 3ware CD in `/packages/cli/linux.`

  Online manual pages are also available in nroff and html formats. These are located in `/packages/cli/tw_cli.8.html` or `tw_cli.8.nroff.`

  You will need to be root or have root privileges to install the CLI to `/usr/sbin` and to run the CLI.

  To install the CLI to a different location, change `/usr/sbin/` to the desired location.

Notes:

The installation location needs to be in the environment path for root to execute the CLI without using complete paths (i.e., if installed to /usr/sbin/, you can type tw_cli on the command line, otherwise you will have to type the complete path: `/home/user/tw_cli`

The 3ware CLI comes in both 32-bit and 64-bit versions. If you are copying the file directly, be sure to copy the correct version for the version of the operating system you are using.

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 7
- "Using a single command with output" on page 7
- "Using an input file to execute a script" on page 8
- "Outputting the CLI to a Text File" on page 9

Examples shown in the CLI Syntax Reference chapter reflect 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 If necessary, change to the directory that contains CLI.

2 Enter the following command:

```
tw_cli
```

(Under Linux, if the directory containing the CLI is not in your path, you may need to type ./tw_cli)

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

```
//localhost>
```

3 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 9690SA-4I4E 4    12      2    0      1       1        -
c1 9650SE-4    4     4      1    0      3       5  TESTING
c2 7500-12    12     8      3    1      2       -        -
```

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 examples

Following is a scripting example for a 4-port controller 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=2-3
/c0 show
```

Following is a scripting example for a 12-port controller using a text file called config_unit.txt, containing three commands. This example sets up a 12-port controller with two units: one with the first 2 drives mirrored, and another with the remaining drives in a RAID 5 array. 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=raid5 disk=2-11
/c0 show
```

To run either of the scripts, enter:

```
tw_cli -f config_unit.txt
```

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 **/c*x*/p*x* 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, **/c*x* 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 **/cx rescan [noscan]**. The brackets [ ] indicate that you may omit the noscan parameter, so that the operation will be reported to the operating system.

Understanding RAID Levels and Concepts

3ware RAID controllers use RAID (Redundant Array of Inexpensive Disks) 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 10
- “Available RAID Configurations” on page 11
- “Determining What RAID Level to Use” on page 17

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 working with a 3ware RAID controller, “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 (RAID 1) 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 (RAID 0) achieve highest transfer rates and performance at the expense of fault tolerance.
- **Distributed Parity**. Parity works in combination with striping on RAID 5, RAID 6, and RAID 50. 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 unit.
- **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

RAID is a method of combining several hard drives into one unit. It offers fault tolerance and higher throughput levels than a single hard drive or group of independent hard drives. RAID levels 0, 1, 10 and 5 are the most popular. AMCC's 3ware controllers support RAID 0, 1, 5, 6, 10, 50, and Single Disk. The information below provides a more in-depth explanation of the different RAID levels.

For how to configure RAID units, see “Configuring a New Unit” on page 85.

RAID 0

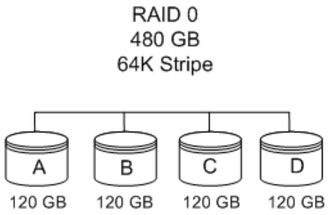
RAID 0 provides improved performance, but no fault tolerance. Since the data is striped across more than one disk, RAID 0 disk arrays achieve high transfer rates because they can read and write data on more than one drive simultaneously. The stripe size is configurable during unit creation. RAID 0 requires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



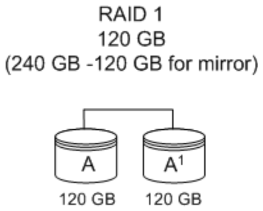
RAID 1

RAID 1 provides fault tolerance and a speed advantage over non-RAID disks. RAID 1 is 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



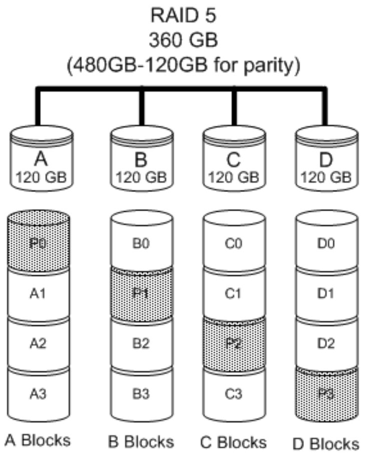
RAID 5

RAID 5 provides performance, fault tolerance, high capacity, and storage efficiency. It requires a minimum of three drives and combines striping data with parity (exclusive OR) to restore data in case of a drive failure. Performance and efficiency increase as the number of drives in a unit increases.

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

RAID 5 is able to tolerate 1 drive failure in the unit.

Figure 3. RAID 5 Configuration Example



RAID 6

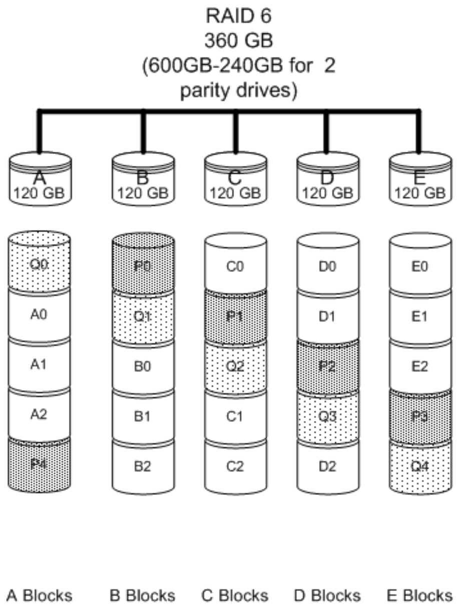
RAID 6 requires a 3ware 9650SE or 9690SA RAID controller.

RAID 6 provides greater redundancy and fault tolerance than RAID 5. It is similar to RAID 5, but has two blocks of parity information (P+Q) distributed across all the drives of a unit, instead of the single block of RAID 5.

Due to the two parities, a RAID 6 unit can tolerate two hard drives failing simultaneously. This also means that a RAID 6 unit may be in two different states at the same time. For example, one sub-unit can be degraded, while another may be rebuilding, or one sub-unit may be initializing, while another is verifying.

AMCC 3ware’s implementation of RAID 6 requires a minimum of five drives. Performance and storage efficiency also increase as the number of drives increase.

Figure 4. RAID 6 Configuration Example



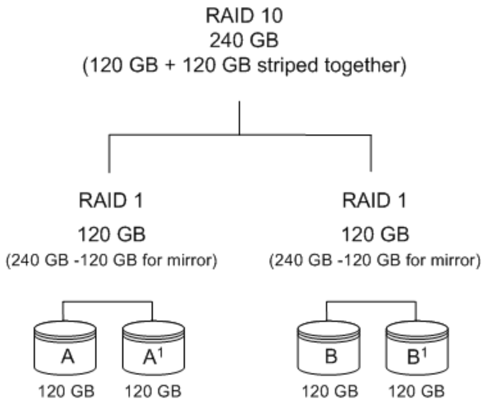
RAID 10

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 5). 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 RAID 1 and RAID 5, since the array can sustain multiple drive failures without data loss. For example, in a twelve-drive RAID 10 array, up to six drives can fail (half of each mirrored pair) and the array will continue to function. 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 5. RAID 10 Configuration Example


RAID 10
240 GB
(120 GB + 120 GB striped together)
RAID 1
120 GB
(240 GB -120 GB for mirror)
RAID 1
120 GB
(240 GB -120 GB for mirror)
A
A1
B
B1
120 GB
120 GB
120 GB
120 GB

RAID 50

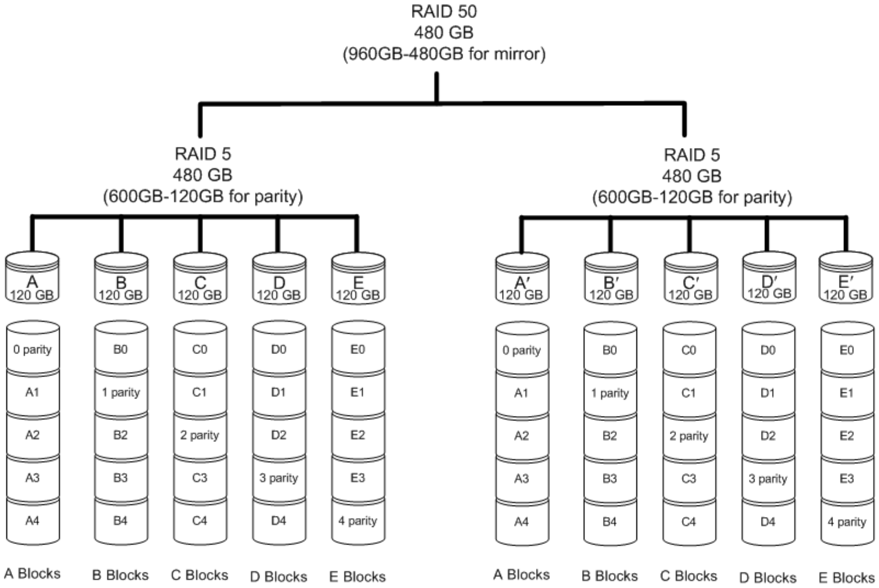
RAID 50 is a combination of RAID 5 with RAID 0. This array type provides fault tolerance and high performance. RAID 50 requires a minimum of six drives.

Several combinations are available with RAID 50. For example, on a 12-port controller, you can have a grouping of 3, 4, or 6 drives. A grouping of 3 means that the RAID 5 arrays used have 3 disks each; four of these 3-drive RAID 5 arrays are striped together to form the 12-drive RAID 50 array. On a 16-port controller, you can have a grouping of 4 or 8 drives.

No more than four RAID 5 subunits are allowed in a RAID 50 unit. For example, a 24-drive RAID 50 unit may have groups of 12, 8, or 6 drives, but not groups of 4 or 3.

In addition, RAID 50 arrays offer a higher degree of fault tolerance than RAID 1 and RAID 5, since the array can sustain multiple drive failures without data loss. For example, in a twelve-drive RAID 50 array, up to one drive in each RAID 5 set can fail and the array will continue to function. Please note that if two or more drives in a RAID 5 set fail, then all of the data will be lost.

Figure 6. RAID 50 Configuration Example



Single Disk

A single drive can be configured as a unit through 3ware software. (3BM, 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 5 units can tolerate one drive failing before losing data.
RAID 6	Provides very high fault tolerance with the ability to protect against two consecutive drive failures. Performance and efficiency increase with higher numbers of drives.
RAID 10	A combination of striped and mirrored units for fault tolerance and high performance.
RAID 50	A combination of RAID 5 and RAID 0. It provides high fault tolerance and 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.

Table 3: Possible Configurations Based on Number of Drives

# Drives	Possible RAID Configurations
1	Single disk or hot spare
2	RAID 0 or RAID 1

Table 3: Possible Configurations Based on Number of Drives

# Drives	Possible RAID Configurations
3	RAID 0 RAID 1 with hot spare RAID 5
4	RAID 5 with hot spare RAID 10 Combination of RAID 0, RAID 1, single disk
5	RAID 6 RAID 5 with hot spare RAID 10 with hot spare Combination of RAID 0, RAID 1, hot spare, single disk
6 or more	RAID 6 RAID 6 with hot spare RAID 50 Combination of RAID 0, 1, 5, 6,10, hot spare, single disk

Using Drive Capacity Efficiently

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

The total unit capacity is defined as follows:

Table 4: 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 6	(number of drives - 2) x (capacity of the smallest drive)
RAID 10	(number of drives / 2) X (capacity of smallest drive)
RAID 50	(number of drives - number of groups of drives) X (capacity of the 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 97.

Note: All drives in a unit must be of the same type, either SAS or SATA.

Support for Over 2 Terabytes

Windows XP (32-bit), Windows 2003 (32-bit and 64-bit without SP1) and Linux 2.4 do not currently recognize unit capacity in excess of 2 TB.

If the combined capacity of the drives to be connected to a unit exceeds 2 Terabytes (TB), you can enable auto-carving when you configure your units.

Auto-carving divides the available unit capacity into multiple chunks of 2 TB or smaller that can be addressed by the operating systems as separate volumes. The carve size is adjustable from 1024 GB to 2048 GB (default) prior to unit creation.

If a unit over 2 TB was created prior to enabling the auto-carve option, its capacity visible to the operating system will still be 2TB; no additional capacity will be registered. To change this, the unit has to be recreated.

For more information, see “Using Auto-Carving for Multi LUN Support” on page 78.

You may also want to refer to Knowledgease article # 13431, at http://www.3ware.com/kb/article.aspx?id=13431.

2

CLI Syntax Reference

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.

Note: The output of some commands varies somewhat for different types of controllers, and may vary if you have an enclosure attached. For most commands where this is the case, examples are provided to show the differences.

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 5: Common Tasks Mapped to CLI Commands

Task	CLI Command	Page
Controller Configuration Tasks		
View information about a controller	/cx show	30
View controller policies	/cx show [attribute] [attribute]	32
Set policies for a controller		
• Modify staggered spinup	/cx set stagger and /cx set spinup	57
• Disable write cache on unit degrade	/cx set ondegrade	57 57
• Enable/disable autocarving	/cx set autocarve	
• Enable/disable autorebuild	/cx set autorebuild	57
• Set the autocarve volume size	/cx set carvesize	57 58

Table 5: Common Tasks Mapped to CLI Commands

Task	CLI Command	Page
Unit Configuration Tasks		
Create a new unit	/cx add	40
Create a hot spare	/cx add	40
Enable/disable unit write cache	/cx/ux set cache	70
Set the queue policy	/cx/ux set qpolicy	72
Set the storsave profile	/cx/ux set storsave	72
Unit Configuration Changes		
Change RAID level	/cx/ux migrate	73
Change stripe size	/cx/ux migrate	73
Expand unit capacity	/cx/ux migrate	73
Delete a unit	/cx/ux del	67
Remove a unit (export)	/cx/ux remove	67
Name a unit	/cx/ux set name	71
Controller Maintenance Tasks		
Update controller with new firmware	/cx update	51
Add a time slot to a rebuild schedule	/cx add rebuild	52
Add a time slot to a verify schedule	/cx add verify	53
Add a time slot to a selftest schedule	/cx add selftest	54
Enable/disable the rebuild/migrate schedule and set the task rate	/cx set rebuild	55
Enable/disable the verify schedule and set the task rate	/cx set verify	56
Enable/disable the selftest schedule	/cx set selftest	56
View Alarms	/cx show alarms	40

Table 5: Common Tasks Mapped to CLI Commands

Task	CLI Command	Page
Unit Maintenance Tasks		
Start a rebuild	/cx/ux start rebuild	68
Start a verify	/cx/ux start verify	68
Pause/resume rebuild	/cx/ux pause rebuild and /cx/ux resume rebuild	69
Stop verify	/cx/ux stop verify	69
Enable/disable autoverify	/cx/ux set autoverify	70
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	85
Check if LED is set to on or off	/cx/px show identify	81
View information for specific drive	/cx/px show	79
View the status of specific drive	/cx/px show status	82
BBU Tasks		
Check on charge and condition of battery	/cx/bbu/ show status	88
Start a test of the battery	/cx/bbu test [quiet]	90
Enclosure Tasks		
View information about an enclosure	/cx/ex show	91
Locate a particular drive slot in an enclosure by blinking an LED	/cx/ex/slotx set identify	95

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 24.

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 29.

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 **/c*x*/u*x* migrate** to change the configuration of a RAID array. For details, see “Unit Object Commands” on page 61.

Port Object Commands. Port object commands provide information and perform actions related to a drive on a specific port or vport. You can 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 79.

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 86.

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 98.

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
commit
flush
rescan
show [attribute [modifier]]
    alarms [reverse]
    diag
    rebuild
    selftest
    ver
    verify
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:
`/c0/u0 show`

If you set the focus to `/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:

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

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

`/` specifies the root

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

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

Example:

```
//localhost> focus /c0/u0
//localhost/c0/u0>

//localhost/c0/u0> focus..
//localhost/c0>

//localhost> focus u0
//localhost/c0/u0>

//localhost/c0> focus /
//localhost>
```

The focus command is available by default. You can disable focus by setting TW_CLI_INPUT_STYLE to **old**. (See “Return Code” on page 103.)

commit

This command sends a commit command to all 3ware controllers in the system. For more information, see “/cx commit” on page 51.

flush

This command sends a flush command to all 3ware controllers in the system. For more information, see “/cx flush” on page 51.

rescan

This command sends a rescan command to all 3ware controllers in the system. For more information, see “/cx rescan [noscan]” on page 50.

show

This command shows a general summary of all detected controllers and enclosures.

The output of this command will vary, depending upon your controller model and whether there is an enclosure with an expander attached.

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 for controller without an enclosure and expander:

Typical output of the Show command for a controller 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*.)

Example for 9690SA-414E with enclosure and expander:

Typical output of the Show command for a system with an enclosure, expander, and a 9690SA-4I4E controller looks like the following:

```
//localhost> show
Ctl    Model       (V)Ports   Drives  Units   NotOpt   RRate   VRate   BBU
---------------------------------------------------------------------------
c0     9690SA-4I4E    12      4       1       0        4       4       -

Encl    Slots   Drives  Fans   TSUnits   PSUnits
---------------------------------------------------
/c0/e0    4       2       1      1         1
```

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 messages at the bottom. The **reverse** attribute displays the most recent message at the top.

show diag

This command shows the diagnostic information of all controllers in the system. The enclosure diagnostic log may be requested by 3ware Customer Support to help troubleshoot problems on your controller.

show rebuild

This command displays all rebuild schedules for the 9000 series 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 52, 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 series 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 54, and see the discussion of background tasks and schedules in *3ware Serial ATA RAID Controller User Guide*.

show ver

This command will show the CLI and API version.

Example:

```
//localhost> show ver
CLI Version = 2.00.03.0xx
API Version = 2.01.00.xx
```

In the above example, “xx” stands for the actual version. See the Release Notes for details.

show verify

This command displays all verify schedules for the 9000 series 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 53, and see the discussion of background tasks and schedules in *3ware Serial ATA RAID Controller User Guide*.

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 51.

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 [pause [lines=n]]
/cx show attribute [attribute ...] where attributes are:
    achip|allunitstatus|
    autocarve(9000 series SX/SE/SA only)|
    autorebuild(9000 series SX/SE/SA only)|bios|
    carvesize(9000 series SX/SE/SA only)|
    ctlbus(9000 series SX/SE/SA only|
    driver|drivestatus[pause[lines=n]]|firmware|
    memory|model|monitor|numdrives|numports|numunits|
    ondegrade(9500S only)|pcb|pchip|serial|
    spinup|stagger|unitstatus|
/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 show phy                       (9690SA only)

/cx add type=<RaidType> disk=<p:-p..> [stripe=<Stripe>]
    [noscan][nocache][group=<3|4|5|6|7|8|9|10|
    11|12|13|14|15|16>]
    [autoverify][noqpolicy][ignorECC][name=string](9000
    series) [v0=n (n=size of first volume in GB)]
    RaidType={raid0,raid1,raid5,raid6(9650SE and higher
    only),raid10,raid50,single,spare}
    [storsave=<protect|balance|perform>]
    (9000 series SX/SE/SA only)
/cx add rebuild=ddd:hh:duration    (9000 series)
/cx add verify=ddd:hh:duration     (9000 series)
/cx add selftest=ddd:hh            (9000 series)

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

/cx set ondegrade=cacheoff|follow  (9500S only)
/cx set spinup=nn                  (9000 series)
```

```
/cx set stagger=nn                  (9000 series)
/cx set autocarve=on|off         (9000 series SX/SE/SA only)
/cx set carvesize=[1024...2048]  (9000 series SX/SE/SA only)
/cx set rebuild=enable|disable|1..5 (9000 series)
/cx set autorebuild=on|off       (9000 series SX/SE/SA only)
/cx set autodetect=on|off disk=<p:-p>|all
/cx set verify=enable|disable|1..5  (9000 series)
/cx set selftest=enable|disable [task=UDMA|SMART](9000
    series)

/cx flush
/cx update fw=filename_with_path [force] (9000 series)
/cx commit          (Windows only. Also known as shutdown)
/cx start mediascan (7000/8000 only)
/cx stop mediascan  (7000/8000 only)
/cx rescan [noscan]
```

/c*x* show

This command shows summary information on the specified controller /c*x*. 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 (or virtual 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 (such as INITIALIZING), %R (percent completion of rebuilding), % V/I/M (percent completion of verifying, initializing, or migrating), stripe size, size (usable capacity) in gigabytes or terabytes, and the auto-verify policy status (on/off)

Possible unit statuses include OK, RECOVERY, INOPERABLE, UNKNOWN, DEGRADED, INITIALIZING, INIT-PAUSED, VERIFYING, VERIFY-PAUSED, REBUILDING, REBUILD-PAUSED, MIGRATING, and MIGRATE-PAUSED. 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 (or virtual 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.

Example output for 9650SE and earlier controllers:

```
//localhost> /c2 show

 Unit  UnitType  Status      %RCmpl  %V/I/M  Stripe  Size(GB)  Cache  AVrfy
 ---------------------------------------------------------------------------
 u0    RAID-5    OK             -       -       64K     596.004   ON     OFF
 u1    RAID-0    OK             -       -       64K     298.002   ON     OFF
 u2    SPARE     OK             -       -       -       149.042   -      OFF
 u3    RAID-1    OK             -       -       -       149.001   ON     OFF

 Port   Status           Unit   Size        Blocks        Serial
 ---------------------------------------------------------------
 p0     OK               u0     149.05 GB   312581808     WD-WCANM1771318
 p1     OK               u0     149.05 GB   312581808     WD-WCANM1757592
 p2     OK               u0     149.05 GB   312581808     WD-WCANM1782201
 p3     OK               u0     149.05 GB   312581808     WD-WCANM1753998
 p4     OK               u2     149.05 GB   312581808     WD-WCANM1766952
 p5     OK               u3     149.05 GB   312581808     WD-WCANM1882472
 p6     OK               u0     149.05 GB   312581808     WD-WCANM1883862
 p7     OK               u3     149.05 GB   312581808     WD-WCANM1778008
 p8     OK               -      149.05 GB   312581808     WD-WCANM1770998
 p9     NOT-PRESENT      -      -           -             -
 p10    OK               u1     149.05 GB   312581808     WD-WCANM1869003
 p11    OK               u1     149.05 GB   312581808     WD-WCANM1762464

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

The above output is for 9650SE and earlier controllers.

Example output for 9690SA controller:

Note that the port information is represented by VPort (virtual port).

```
 Unit  UnitType  Status        %RCmpl  %V/I/M  Stripe  Size(GB)  Cache  AVrfy
 ----------------------------------------------------------------------------
 u0    SPARE     OK              -       -       -       149.042   -      OFF
 u1    Single    OK              -       -       -       149.051   OFF    OFF
 VPort Status       Unit    Size    Type  Phy Encl-Slot    Model
 ----------------------------------------------------------------------------
 p0    OK             -    149.05 GB SATA  3   -       WDC WD1600JS-22NCB1
 p1    OK            u0    149.05 GB SATA  0   -       WDC WD1600JS-22NCB1
 p2    OK            u1    149.05 GB SATA  2   -       WDC WD1600JS-22NCB1
 p3    OK             -    34.18 GB  SAS   6   -        SEAGATE ST936701SS
```

/c*x* 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 SX/SE/SA only), autorebuild (9000 series SX/SE/SA only), bios, carvesize (9000 series SX/SE/SA only), driver, drivestatus, firmware, memory, model, monitor, numdrives, numports, numunits, ctlbus (9000 series SX/SE/SA 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 3.x
```

(In the sample output above, “x” will be replaced with the actual version number.)

/c*x* show achip

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

Example:

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

/c*x* show allunitstatus

This command presents a count of total and Not Optimal units managed by the specified controller /c*x*. For more about the meaning of Not Optimal, see “Shell Object Commands” on page 24.

Example:

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

/c*x* show autocarve

This feature only applies to 9000 series SX/SE/SA 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 memory”, 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
```

/c*x* show autorebuild

This feature only applies to 9000 series SX/SE/SA 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 58.

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
```

/c*x* show bios

This command reports the BIOS version of controller /c*x*.

Example:

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

/c*x* show carvesize

This feature only applies to 9000 series SX/SE/SA model 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
```

/c*x* show ctlbus

This feature only applies to 9000 series SX/SE/SA model controllers.

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

Example for 9690SA:

```
 //localhost> /c2 show ctlbus
/c2 Controller Bus Type = PCIe
/c2 Controller Bus Width = 8 lanes
/c2 Controller Bus Speed = 2.5 Gbps/lane
```

/c*x* show driver

This command reports the device driver version associated with controller /c*x*.

Example:

```
 //localhost> /c0 show driver
 /c0 Driver Version = 3.x
```

/c*x* 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 for 9650SE and earlier controllers:

```
 //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      -      -           -             -
```

Example for 9690SA controller:

```
 //localhost> /c2 show drivestatus
VPort Status     Unit Size      Type  Phy Encl-Slot    Model
---------------------------------------------------------------------------
p0    OK         u0   34.25 GB  SAS   -   /c2/e0/slt0  MAXTOR ATLAS15K2_36
p1    OK         u0   34.25 GB  SAS   -   /c2/e0/slt1  MAXTOR ATLAS15K2_36
p2    OK         u0   34.25 GB  SAS   -   /c2/e0/slt2  MAXTOR ATLAS15K2_36
p3    OK         u0   34.18 GB  SAS   -   /c2/e1/slt0  HITACHI HUS151436VL
p4    OK         u0   34.18 GB  SAS   -   /c2/e1/slt1  HITACHI HUS151436VL
p5    OK         u0   34.18 GB  SAS   -   /c2/e1/slt2  HITACHI HUS151436VL
p6    OK         u0   34.25 GB  SAS   -   /c2/e0/slt3  MAXTOR ATLAS15K2_36
p7    OK         u0   34.25 GB  SAS   -   /c2/e0/slt4  MAXTOR ATLAS15K2_36
p8    OK         u0   34.25 GB  SAS   -   /c2/e0/slt5  MAXTOR ATLAS15K2_36
p9    OK         u0   34.25 GB  SAS   -   /c2/e0/slt6  MAXTOR ATLAS15K2_36
p10   OK         u0   34.18 GB  SAS   -   /c2/e1/slt3  HITACHI HUS151436VL
p11   OK         u0   34.18 GB  SAS   -   /c2/e1/slt4  HITACHI HUS151436VL
p12   OK         u0   34.18 GB  SAS   -   /c2/e1/slt5  HITACHI HUS151436VL
p13   OK         u0   34.18 GB  SAS   -   /c2/e1/slt6  HITACHI HUS151436VL
p14   OK         u0   34.25 GB  SAS   -   /c2/e0/slt7  MAXTOR ATLAS15K2_36
p15   OK         u0   34.25 GB  SAS   -   /c2/e0/slt8  MAXTOR ATLAS15K2_36
p16   OK         u0   34.25 GB  SAS   -   /c2/e0/slt9  MAXTOR ATLAS15K2_36
p17   OK         u0   34.25 GB  SAS   -   /c2/e0/slt10 MAXTOR ATLAS15K2_36
p18   OK         u0   34.18 GB  SAS   -   /c2/e1/slt7  HITACHI HUS151436VL
p19   OK         u0   34.18 GB  SAS   -   /c2/e1/slt8  HITACHI HUS151436VL
p20   OK         u0   34.18 GB  SAS   -   /c2/e1/slt9  HITACHI HUS151436VL
p21   OK         u0   34.18 GB  SAS   -   /c2/e1/slt10 HITACHI HUS151436VL
p22   OK         u0   34.25 GB  SAS   -   /c2/e0/slt11 MAXTOR ATLAS15K2_36
p23   OK         u0   34.25 GB  SAS   -   /c2/e0/slt12 MAXTOR ATLAS15K2_36
p24   OK         -    34.25 GB  SAS   -   /c2/e0/slt13 MAXTOR ATLAS15K2_36
p25   OK         -    34.25 GB  SAS   -   /c2/e0/slt14 MAXTOR ATLAS15K2_36
p26   OK         -    34.18 GB  SAS   -   /c2/e1/slt11 HITACHI HUS151436VL
p27   OK         -    34.18 GB  SAS   -   /c2/e1/slt12 HITACHI HUS151436VL
p28   OK         -    34.18 GB  SAS   -   /c2/e1/slt13 HITACHI HUS151436VL
p29   OK         -    34.18 GB  SAS   -   /c2/e1/slt14 HITACHI HUS151436VL
p30   OK         -    34.25 GB  SAS   -   /c2/e0/slt15 MAXTOR ATLAS15K2_36
p31   OK         -    34.18 GB  SAS   -   /c2/e1/slt15 HITACHI HUS151436VL
```

/c*x* show firmware

This command reports the firmware version of controller /c*x*.

Example:

```
 //localhost> /c0 show firmware
 /c0 Firmware Version = FE9X 3.03.06.X03
```

/*cx* show memory

This command reports the available memory on the controller.

Note: Some memory is reserved for use by the controller, so the amount of memory available will be less than the controller actually has installed. The 9690SA controller has 512MB of memory of which 448MB is available.

Example:

```
//localhost> /c2 show memory
/c2 Available Memory = 448MB
```

/*cx* show model

This command reports the controller model of controller /*cx*.

Example:

```
//localhost> /c0 show model
/c0 Model = 9690SA-8E
```

/*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
```

c*x* 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.

On 9500S and earlier controllers, physically-removed disk(s) will still be counted. For a workaround, see “/cx/px show smart” on page 82.

Example:

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

/*cx* show numports

For controllers earlier than the 9690SA, 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
```

For the 9690SA, this command reports the number of virtual ports being used.

Example for 9690SA:

```
//localhost> /c2 show numports
/c2 Number of Ports = 32
```

/*cx* show numunits

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

Example:

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

/*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 pcb

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

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 /c*x*.

Example:

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

This feature only applies to 9000 series controllers.

This command reports the number of concurrent SAS and SATA disks that will spin up when the system is powered on, after waiting for the number of seconds specified with the `set stagger` command. Spinup does not work with SAS or SATA disks attached to an expander.

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. Spinup does not work with SAS or SATA disks attached to an expander.

Example:

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

/*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> /c2 show unitstatus

  Unit  UnitType  Status     %RCmpl  %V/I/M  Stripe  Size(GB)  Cache  AVrfy
  ------------------------------------------------------------------------
  u0    RAID-5    OK           -       -       64K     596.004   ON     OFF
  u1    RAID-0    OK           -       -       64K     298.002   ON     OFF
  u2    SPARE     OK           -       -       -       149.042   -      OFF
  u3    RAID-1    OK           -       -       -       149.001   ON     OFF
```

/c*x* show all

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

Example:

```
//localhost>> /c0 show all

/c0 Driver Version = 3.00.03.058
/c0 Model = 9690SA-8E
/c0 Available Memory = 448MB
/c0 Firmware Version = FH9X 4.03.00.012
/c0 Bios Version = BE9X 4.01.00.008
/c0 Boot Loader Version = BL9X 3.08.00.001
/c0 Serial Number = 3ware Internal Use
/c0 PCB Version = Rev 032
/c0 PCHIP Version = 2.00
/c0 ACHIP Version = 1501290C
/c0 Controller Phys = 8
/c0 Connections = 10 of 128
/c0 Drives = 10 of 128
/c0 Units = 2 of 128
/c0 Active Drives = 7 of 128
/c0 Active Units = 2 of 32
/c0 Max Drives Per Unit = 32
/c0 Total Optimal Units = 2
/c0 Not Optimal Units = 0
/c0 Disk Spinup Policy = 1
/c0 Spinup Stagger Time Policy (sec) = 1
/c0 Auto-Carving Policy = off
/c0 Auto-Carving Size = 2048 GB
/c0 Auto-Rebuild Policy = off
/c0 Controller Bus Type = PCIe
/c0 Controller Bus Width = 4 lanes
/c0 Controller Bus Speed = 2.5 Gbps/lane

Unit  UnitType  Status       %RCmpl  %V/I/M  Stripe  Size(GB)  Cache  AVrfy
---------------------------------------------------------------------------
--
u0    RAID-5    OK            -       -       64K     134.07    ON     OFF
u1    RAID-0    OK            -       -       64K     67.0352   ON     OFF

VPort Status       Unit Size      Type  Phy Encl-Slot    Model
---------------------------------------------------------------------------
p0    OK           u0   68.36 GB  SAS   -   /c0/e0/slt0  HITACHI HUS151473VL
p1    OK           u0   34.25 GB  SAS   -   /c0/e0/slt2  MAXTOR ATLAS15K2_36
p2    OK           u0   34.25 GB  SAS   -   /c0/e0/slt3  MAXTOR ATLAS15K2_36
p3    OK           u0   34.24 GB  SAS   -   /c0/e0/slt4  FUJITSU MAX3036RC
```

```
p4    OK           u0   34.25 GB  SAS   -   /c0/e0/slt5  MAXTOR ATLAS15K2_36
p5    OK           u1   34.25 GB  SAS   -   /c0/e0/slt6  MAXTOR ATLAS15K2_36
p6    OK           u1   68.36 GB  SAS   -   /c0/e0/slt7  HITACHI HUS151473VL
p7    OK           -    34.25 GB  SAS   -   /c0/e0/slt8  MAXTOR ATLAS15K2_36
p8    OK           -    34.18 GB  SAS   -   /c0/e0/slt9  HITACHI HUS151436VL
p9    OK           -    34.25 GB  SAS   -   /c0/e0/slt10 MAXTOR ATLAS15K2_36
```

/c*x* 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 at the bottom. The user can also use the [**reverse**] attribute to display the most recent alarm or AEN message at the top.

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> /c1 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
```

/c*x* 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
```

/c*x* show phy

This command is for the 9690SA only. It reports a list of the phys for the controller.

Example of 9690SA-8E connected to drives in an enclosure:

```
                             Device            --- Link Speed (Gbps) ---
Phy     SAS Addesss          Type     Device   Supported  Enabled  Limit
--------------------------------------------------------------------------
phy0    500050e000030232     ENCL     N/A      1.5-3.0    3.0      Auto
phy1    500050e000030232     ENCL     N/A      1.5-3.0    3.0      Auto
phy2    500050e000030232     ENCL     N/A      1.5-3.0    3.0      Auto
phy3    500050e000030232     ENCL     N/A      1.5-3.0    3.0      Auto
phy4    500050e000030236     ENCL     N/A      1.5-3.0    3.0      Auto
phy5    500050e000030236     ENCL     N/A      1.5-3.0    3.0      Auto
phy6    500050e000030236     ENCL     N/A      1.5-3.0    3.0      Auto
phy7    500050e000030236     ENCL     N/A      1.5-3.0    3.0      Auto
```

Example of 9690SA-8I with directly attached drives:

```
                             Device            --- Link Speed (Gbps) ---
Phy     SAS Addesss          Type     Device   Supported  Enabled  Limit
--------------------------------------------------------------------------
phy0    500050e000030442     SAS      0        1.5-3.0    3.0      Auto
phy1    500050e000030442     SAS      2        1.5-3.0    3.0      Auto
phy2    500050e000030442     SAS      4        1.5-3.0    3.0      Auto
phy3    500050e000030442     SAS      6        1.5-3.0    3.0      Auto
phy4    500050e000030446     SAS      1        1.5-3.0    3.0      Auto
phy5    500050e000030446     SATA     3        1.5-3.0    1.5      Auto
phy6    500050e000030446     SATA     5        1.5-3.0    1.5      Auto
phy7    500050e000030446     SATA     7        1.5-3.0    3.0      Auto
```

/c*x* 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 52.

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 for 9650SE controller:

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:

```
$ tw_cli /c1 show

Unit  UnitType  Status       %RCmpl %V/I/M  Stripe  Size(GB)  Cache  AVrfy
--------------------------------------------------------------------------
u0    RAID-5    INITIALIZING   0      -      64K    521.466    ON     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

Name  OnlineState  BBUReady  Status    Volt     Temp    Hours  LastCapTest
--------------------------------------------------------------------------
bbu   On           Yes       OK        OK       OK      0      xx-xxx-xxxx
```

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> /c1 set rebuild=enable
Enabling scheduled rebuilds on controller /c1 ...Done.

//localhost> /c1 show rebuild
Rebuild Schedule for Controller /c1
========================================================
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

$ tw_cli /c1 show

Unit  UnitType  Status        %RCmpl %V/I/M  Stripe  Size(GB)  Cache  AVrfy
--------------------------------------------------------------------------
u0    RAID-5    INIT-PAUSED  0       -       64K     521.466   ON     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

Name  OnlineState  BBUReady  Status    Volt     Temp     Hours  LastCapTest
---------------------------------------------------------------------------
bbu   On           Yes       OK        OK       OK       0      xx-xxx-xxxx
```

/c*x* 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 54.

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.

Note: UDMA mode is applicable only for PATA (parallel ATA) drives on earlier 3ware controllers. It is not applicable for SATA or SAS drives.

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.

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. (SMART data is available only for SATA drives.)

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> /c1 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
```

/c*x* 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 53.

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 and RAID 6, 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, 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.

/c*x* add type=*<RaidType>* disk=*<p:-p>* [stripe=*Stripe*] [noscan] [group=*<3|4|5|6|7|8|9|10|11|12/13|14|15|16>*] [nocache] [autoverify] [noqpolicy][ignoreECC] [name=*string*] [storsave=<protect|balance|perform>] [v0=n]

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, a storsave policy of protect will be set, and the drive queuing policy is enabled. 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.

Note: By default, write cache is enabled. However, if the controller does not have a BBU installed, a message will warn you that you could lose data in the event of a power failure.

Enabling write cache will improve write performance greatly, but you are at risk of losing data if a power failure occurs when data is still in the cache. You may want to obtain a UPS to safeguard against power loss.

/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, raid6 (9650SE and higher only), raid10, raid50, single, and spare.

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.

The following table shows supported types and controller models.

Table 6: Supported RAID Types

Model	R0	R1	R5	R6	R10	R50	Single	Spare
7K/8K	Yes	Yes	Yes	No	Yes	No	No	Yes
9000[a]	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
9650SE and 9690SA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

[a] Models 9500S, 9550SX, and 9590SE

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.

If you have a 9690SA controller, the syntax is the same even though you are technically addressing vports.

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 my migrating the unit.

Table 7: Supported Stripe Sizes (KB)

Model	R0	R1	R5	R6	R10	Spare	R50	Single
7K/8K	64	N/A	64	N/S	64	N/A	N/S	N/S
	128				128			
	256				256			
	512				512			
	1024				1024			
9000[a]	16	N/A	16	N/S	16	N/A	16	N/A
	64		64		64		64	
	256		256		256		256	

Table 7: Supported Stripe Sizes (KB)

Model	R0	R1	R5	R6	R10	Spare	R50	Single
9650SE and 9690SA	16	N/A	16		16	N/A	16	N/A
	64		64	64	64		64	
	256		256		256		256	

a. Models 9500S, 9550SX, and 9590SE

group=*3|4|5|6|7|8|9|10|11|12/13|14|15|16* 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 through 16. However, no more than 4 RAID 5 subunits are allowed in a RAID 50 unit.

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 disks can be grouped into groups of 3, 4, or 6 drives (the group of 4 drives is set by default as it provides the best of net capacity and performance). For 15 disks, the disks can be grouped into 5 drives (3 drive groups would make 5 subunits, you can have a maximum of 4 subunits). For 16 disks, the disks can be grouped into groups of 4 or 8 drives.

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 (uninterruptible 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 70. This feature is not supported on model 7000/8000.

noqpolicy attribute instructs CLI to disable the qpolicy (drive queuing for SATA drives only) on the newly created unit. The default is for the qpolicy to be on (in other words, noqpolicy is not specified). For a spare unit, drive queuing is not meaningful, so the noqpolicy cannot be set. During unit creation, specifying ***noqpolicy*** for a spare returns an error. (If the spare unit becomes a true unit, it will adopt the qpolicy of the "new" unit.) For more about drive queuing, see "/cx/ux show qpolicy" on page 64 and "/cx/ux set qpolicy=on|off" on page 72.

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 71. 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 8: Supported Model-Unit Types for ignoreECC

Model	R-0	R-1	R-5	R-6	R-10	R-50	Single	Spare
7K/8K	No	No	No	N/A	No	No	No	No
9000[a]	No	Yes	Yes	N/A	Yes	Yes	No	No
9650SE	No	Yes	Yes	Yes	Yes	Yes	No	No

[a] Models 9500S, 9550SX, and 9590SE

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 71* and "/cx/ux show name" on page 64.

storsave=protect|balance|perform attribute allows user to set the storsave policy of the new unit. This feature is only for 9000 series SX/SE/SA controllers. For more information, see "/cx/ux set storsave=protect|balance|perform [quiet]" on page 72.

v0=n attribute allows the user to set the size of the first volume of the new unit. The first volume may be, but is not required to be, the boot LUN. The input is a positive integer n in units of gigabytes (GB). If the input is less than or equal to zero, the unit is created with the first volume ``uncarved", that is, the size would be the entire array size.

The following example shows a RAID 5 unit being created with the first volume size set to 10 GB.

```
//localhost> /c0 add type=raid5 disk=2-5 v0=10
```

```
 Creating new unit on Controller /c0 ... Done. The new unit is /
c0/u0.
 Setting write cache=ON for the new unit ... Done.
 Setting default Command Queuing Policy for unit /c0/u0 to [on] 
... Done.
```

After the unit creation, a subsequent ``show" command for the unit shows the the volume size(s):

//localhost> /c0/u0 show

```
Unit     UnitType  Status         %RCmpl  %V/I/M  VPort  Stripe  Size(GB)
------------------------------------------------------------------------
u0       RAID-5    OK             -       -       -      64K     1117.56
u0-0     DISK      OK             -       -       p2     -       372.519
u0-1     DISK      OK             -       -       p3     -       372.519
u0-2     DISK      OK             -       -       p4     -       372.519
u0-3     DISK      OK             -       -       p5     -       372.519
u0/v0    Volume    -              -       -       -      -       10
u0/v1    Volume    -              -       -       -      -       1107.56
```

/c*x* rescan [noscan]

This command instructs the controller to rescan all ports, vports, and phys and reconstitute all units. The controller will update its list of 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 bays are required unless you first power-down the system to prevent system hangs and electrical damage.

Example:

```
 //localhost> /c1 rescan
 Rescanning controller /c1 for units and drives ...Done
 Found following unit(s): [/c1/u3]
 Found following drive(s): [/c1/p7, /c1/p8]
```

/c*x* 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 file systems do not require this command since they have their own ways of notifying the controller to do clean up for shut down.

/c*x* flush

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

/c*x* 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.

Important: Before you update the firmware on your controller, please follow these recommendations:

1) Back up your data. Updating the firmware can render the device driver and/or management tools incompatible.

2) Make sure you have a copy of the current firmware image so that you can roll back to it, if required.

3) Close all applications before beginning the update of the firmware.

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.

Note: The prom image number will vary for different controllers.

Example:

```
 //localhost> /c0 update fw=/tmp/prom0006.img
Warning: Updating the firmware can render the device driver and/or management
tools incompatible. Before you update the firmware, it is recommended that
you:

1) Back up your data.
2) Make sure you have a copy of the current firmware image so that you can
roll back, if necessary.
3) Close all applications.

 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.

/c*x* 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 **/c*x* add rebuild**.

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

1 Use the **/c*x* 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 **/c*x* del rebuild** command to remove them. (For details, see page 55.)

3 Use the **/c*x* add rebuild** command to create the rebuild schedule slots you want (described above.)

4 Use the **/c*x* set rebuild=enable** command to enable the schedule (this enables all rebuild schedule slots). (For details, see page 55.)

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

/c*x* 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> /c1 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 **/c*x* show verify** command to display the current schedule for verify tasks. (For details, see page 45.)

2 If any of the scheduled tasks do not match your desired schedule, use the **/c*x* del verify** command to remove them. (For details, see page 55.)

3 Use the **/c*x* 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 56.)

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 70.)

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

/c*x* 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 52.)

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 55.)

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 56.)

/c*x* 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

/c*x* del verify=*slot_id*

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

Example:

```
//localhost> /c1 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

/c*x* del selftest=*slot_id*

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

Example:

```
//localhost> /c1 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.

/c*x* set rebuild=enable|disable|1..5

This command enables or disables all rebuild background task slots on controller **/c*x*** 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 series controllers, 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.

/c*x* set verify=enable|disable|1..5

This command enables or disables all verify background task slots on controller **/c*x*** 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 series controllerss, 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 70.

/c*x* 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 series controllers. 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.

/c*x* 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 70.)

/c*x* 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 (**/c*x* 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.

/c*x* 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.

/c*x* set autocarve=on|off

This feature only applies to 9000 series SX/SE/SA model 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.
```

/c*x* set carvesize=[1024..2048]

This feature only applies to 9000 series SX/SE/SA model 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.
```

/c*x* set autorebuild=on|off

This feature only applies to 9000 series SX/SE/SA model 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 to cause a drive timeout error and 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 set autodetect=on|off disk=<p:-p>|all

This command is only for 9000 series controllers.

This command is associated with the staggered spin-up feature when hot-swapping drives. When staggered spin-up is enabled (see command /cx set spinup and /cx set stagger), during a reset or power on, the controller will spin up all detected drives with a delay between each spinup, allowing the spread of power consumption on the power supply. When a drive is hot-swapped, (as opposed to when it has just been powered on or reset), the default behavior of the system is to automatically detect and immediately spin up the drives. This command can change the default behavior and set the controller to do a staggered spinup for hot-swapped drives.

Note: The autodetect setting cannot be shown in CLI or displayed in 3DM or 3BM. This feature may be added in a future release.

autodetect=on|off enables or disables automatic detection of drives on the controller’s ports for staggered spin-up.

disk=<p:-p>|all specifies one or many disks (that is, drives, ports, or vports). If a port is empty (no drive is inserted), the echo message of the command refers to a port. If there is already a drive inserted, the message refers to a disk. The example below shows that autodetect has been set to ***off*** to initiate staggered spin-up during hot-swapping, where port 3 was empty and ports 5 and 6 had drives inserted.

Example:

```
//localhost>> /c0 set autodetect=off disk=3:5-6
Setting Auto-Detect on /c0 to [off] for port [3] and for disk
[5,6]... Done
```

If “disk=all,” then all of the drives or ports for that controller are specified. For the 9690SA controller, it would spinup all directly attached SAS and SATA drives, but not any drives attached to an expander.

Example:

```
//localhost>> /c0 set autodetect=off disk=all
Setting Auto-Detect on /c2 to [off] for all disks/ports... Done.
```

Usage Scenario:

If you are hot-plugging a large number of drives at the same time and are concerned that you might overload the power supply, you might use this command as follows:

1 Issue the command (set autodetect=off) to disable automatic detection of the ports for staggered spin-up.

2 If the ports are not empty, pull the drives out of the specified ports.

3 Insert (or replace) the drives at the ports specified.

4 Issue the command (set autodetect=on) to enable auto detect of the ports with the newly inserted drives.

The preceding steps would spin up the newly inserted drives in a staggered manner. Please note that the command takes longer for ports that do not have drives inserted, since the controller allows time for the empty ports to respond.

/c*x* 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*.

/c*x* stop mediascan

This command applies only to 7000/8000 controllers.

This commands 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:
    autoverify (9000 series)|
    initializestatus|cache|name(9000 series)|
    qpolicy(9000 series SX/SE/SA only)|rebuildstatus|
    serial(9000 series)|status|verifystatus|
    storsave(9000 series SX/SE/SA only)
    |volumes(9000 series)|
    ignoreECC (9000 series)|identify (9000 series SX/SE/SA
    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    (9000 series SX/SE/SA only))
/cx/ux set ignoreECC=on|off
/cx/ux set qpolicy=on|off     (9000 series SX/SE/SA only)
/cx/ux set name=string         (9000 series)
/cx/ux set storsave=protect|balance|perform [quiet](9000
    series SX/SE/SA only)
/cx/ux migrate type=RaidType [disk=p:-p]
    [group=3|4|5|6|7|8|9|10|11|12|13|14|15|16]
    [stripe=Stripe] [noscan] [nocache] [autoverify]
    (9000 series) RaidType = {raid0, raid1, raid5,
    raid6(9650SE and later only), raid10, raid50, single}
/cx/ux remove [noscan] [quiet]
```

/c*x*/u*x* show

This command shows summary information about the specified unit **/c*x*/u*x***. 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. Similarly, if the unit was created using the 3ware BIOS utility 3BM and a size was entered in the Boot Volume Size field, multiple volumes were created and will be displayed. Note that a volume created using the Boot Volume Size feature does not have to be used as a boot volume.

One application of the **/c*x*/u*x* 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 for 9650SE and earlier controllers:

```
//localhost> /c0/u0 show

 Unit     UnitType  Status         %RCmpl  %V/I/M  Port  Stripe  Size(GB)
 ------------------------------------------------------------------------
 u0       RAID-50   OK             -       -       -     64K     596.05
 u0-0     RAID-5    OK             -       -       -     64K     -
 u0-0-0   DISK      OK             -       -       p0    -       149.10
 u0-0-1   DISK      OK             -       -       p2    -       149.10
 u0-0-2   DISK      OK             -       -       p3    -       149.10
 u0-1     RAID-5    OK             -       -       -     64K     -
 u0-1-0   DISK      OK             -       -       p4    -       149.10
 u0-1-1   DISK      OK             -       -       p5    -       149.10
 u0-1-2   DISK      OK             -       -       p6    -       149.10
```

Example for 9690SAcontrollers:

```
 //localhost> /c0/u1 show

 Unit     UnitType  Status         %RCmpl  %V/I/M  VPort  Stripe  Size(GB)
 ------------------------------------------------------------------------
 u1       RAID-0    OK             -       -       -     64K     3576.06
 u1-0     DISK      OK             -       -       p0    -       298.01
 u1-1     DISK      OK             -       -       p1    -       298.01
 u1-2     DISK      OK             -       -       p2    -       298.01
 u1-3     DISK      OK             -       -       p3    -       298.01
 u1-4     DISK      OK             -       -       p4    -       298.01
 u1-5     DISK      OK             -       -       p5    -       298.01
 u1-6     DISK      OK             -       -       p6    -       298.01
 u1-7     DISK      OK             -       -       p7    -       298.01
 u1-8     DISK      OK             -       -       p8    -       298.01
 u1-9     DISK      OK             -       -       p9    -       298.01
 u1-10    DISK      OK             -       -       p10   -       298.01
 u1-11    DISK      OK             -       -       p11   -       298.01
 u1/v0    Volume    -              -       -       -     -       2047.00
 u1/v1    Volume    -              -       -       -     -       1529.06
```

/c*x*/u*x* 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), autoverify (9000 series), cache, ignoreECC (9000 series), identify (9000 series SX/SE/SA only), qpolicy (9000 series SX/SE/SA only), rebuildstatus, serial (9000 series), status, storsave (9000 series SX/SE/SA only), verifystatus, and volumes (9000 series).

/c*x*/u*x* 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
```

/c*x*/u*x* 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
```

/c*x*/u*x* show identify

This feature only applies to 9000 series SX/SE/SA model controllers.

This command is related to the **/c*x*/u*x* 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
```

/c*x*/u*x* 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
```

/c*x*/u*x* 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
```

/c*x*/u*x* 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
```

/c*x*/u*x* show qpolicy

This feature only applies to 9000 series SX/SE/SA model controllers.

This command reports the queue policy of the firmware for SATA drives. Qpolicy is not applicable to SAS drives. If the queue policy is on, the firmware utilizes the drive’s queueing policy. If any drives do not support a queueing policy, this policy will have no effect on those drives.

For a spare unit, drive queuing is not meaningful or applicable. When a spare becomes part of a true unit during a rebuild, it will adopt the queue policy of the "new" parent unit. Thus, this command does not show the queue policy for the spare unit type.

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

Note that queuing information is not available for SAS drives.

Example:

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

/c*x*/u*x* 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*/u*x* 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*/u*x* 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*/u*x* show storsave

This feature only applies to 9000 series SX/SE/SA 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 72.

Example:

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

/*cx*/u*x* 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
```

/c*x*/u*x* 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 auto-carving is enabled and/ or a boot LUN was specified.

Example:

```
//localhost> /c0/u0 show volumes
/c0/u0 volume(s) = 1
```

/c*x*/u*x* 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. Similarly, if the unit was created using the 3ware BIOS utility 3BM and a size was entered in the Boot Volume Size field, multiple volumes were created and will be displayed. Note that a volume created using the Boot Volume Size feature does not have to be used as a boot volume.

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 Cache State = on
 /c0/u1 volume(s) = 2
 /c0/u1 name = myarray
 /c0/u1 serial number = C6CPR7JMF98DA8001DF0
 /c0/u1 Ignore ECC policy = on
 /c0/u1 Auto Verify Policy = on
 /co/u1 Storsave policy = protection
 /c0/u1 Command Queuing Policy = on

 Unit     UnitType  Status         %RCmpl  %V/I/M  VPort  Stripe  Size(GB)
 ------------------------------------------------------------------------
 u1       RAID-0    OK             -       -       -     64K     3576.06
 u1-0     DISK      OK             -       -       p0    -       298.01
 u1-1     DISK      OK             -       -       p1    -       298.01
 u1-2     DISK      OK             -       -       p2    -       298.01
 u1-3     DISK      OK             -       -       p3    -       298.01
 u1-4     DISK      OK             -       -       p4    -       298.01
 u1-5     DISK      OK             -       -       p5    -       298.01
 u1-6     DISK      OK             -       -       p6    -       298.01
 u1-7     DISK      OK             -       -       p7    -       298.01
 u1-8     DISK      OK             -       -       p8    -       298.01
 u1-9     DISK      OK             -       -       p9    -       298.01
 u1-10    DISK      OK             -       -       p10   -       298.01
```

```
u1-11    DISK      OK             -       -       p11   -       298.01
u1/v0    Volume    -              -       -       -     -       2047.00
u1/v1    Volume    -              -       -       -     -       1529.06
```

/c*x*/u*x* remove [noscan] [quiet]

This command allows you to remove (previously called “export”) a unit. Removing a unit instructs the firmware to remove the specified unit from its poll of managed units, but retains the DCB (Disk Configuration Block) metadata. A removed unit can be moved to a different controller.

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 remove

```
Removing /c0/u0 will take the unit offline.
Do you want to continue?
Y|N [N]:
```

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

/c*x*/u*x* 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
Deleting /c0/u0 will cause the data on the unit to be
permanently lost.
Do you want to continue ? Y|N [N]:
```

/c*x*/u*x* 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 6, RAID 10, and RAID 50.

During rebuild, bad sectors on the source disk will cause the rebuild to fail. RAID 6 arrays are less susceptible to failing since two copies of the data exist. 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 84 for details. Also refer to the command "/cx rescan [noscan]" on page 50 to add new drives or to retry the original drive.

If you are rebuilding a RAID 50, RAID 6, 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.

/c*x*/u*x* start verify

This command starts a background verification process on the specified unit **/c*x*/u*x***. 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 9: Supported RAID (Logical Unit) Types for Verification

Model	R0	R1	R5	R6	R10	R50	Single	Spare
7K/8K	No	Yes	Yes	N/A	Yes	N/A	N/A	No

Table 9: Supported RAID (Logical Unit) Types for Verification

Model	R0	R1	R5	R6	R10	R50	Single	Spare
9000[a]	Yes	Yes	Yes	N/A	Yes	Yes	Yes	Yes
9650SE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

[a] Models 9500S, 9550SX, and 9590SE

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.

/c*x*/u*x* pause rebuild

This command allows you to pause the rebuild operation on the specified unit **/c*x*/u*x***.

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 **/c*x*/u*x* 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.

/c*x*/u*x* resume rebuild

This command allows you to resume the rebuild operation on the specified unit **/c*x*/u*x***.

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 /**c*x*/u*x* 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.

/c*x*/u*x* stop verify

This command stops a background verification process on the specified unit **/c*x*/u*x***. Table 9 on page 68 shows the supported matrix as a function of the controller model and logical unit type.

/c*x*/u*x* flush

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

/c*x*/u*x* 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 **/c*x*/u*x*** 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 53.

/c*x*/u*x* set cache=on|off [quiet]

This command allows you to turn **on** or **off** the write cache for a specified unit **/c*x*/u*x***. This feature is supported on all controllers.

By default, cache is **on**.

Write cache includes the disk drive cache and controller cache.

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 10: Supported RAID Types for Caching

Model	R0	R1	R5	R6	R10	R50	Single	Spare
7K/8K	Yes	Yes	Yes	N/A	Yes	N/A	N/A	No
9000[a]	Yes	Yes	Yes	N/A	Yes	Yes	Yes	No
9650SE and 9690SA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No

[a] Models 9500S, 9550SX, and 9590SE

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

/c*x*/u*x* set identify=on|off

This feature only applies to 9000 series SX/SE/SA model controllers.

This feature requires an enclosure that uses an integrated AMCC/3ware CCU (chassis control unit), such as the 3ware Sidecar, or a chassis from a third-party vendor. (Check the 3ware web site for a list of chassis vendors that support enclosure services, as they become available.) For additional information about enclosure-related commands, see “Enclosure Object Commands” on page 91.

This command allows you to identify a unit within an enclosure by blinking the LEDs associated with the drive slots of the specified unit.

Example:

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

/c*x*/u*x* 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**.

See Table 8, “Supported Model-Unit Types for ignoreECC,” on page 49

/c*x*/u*x* 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.

/c*x*/u*x* set qpolicy=on|off

This feature only applies to 9000 series SX/SE/SA model controllers.

This command sets the queue policy of the firmware. If the queue policy is **on**, the firmware utilizes the drive’s queueing policy. If any drives do not support a queueing policy, this policy will have no effect on those drives.

For a spare unit, drive queuing is not meaningful or applicable. When a spare becomes part of a true unit during a rebuild, it adopts the queue policy of the "new" parent unit. Thus, this command does not set the queue policy for the spare unit type.

Note: Queing is only applicable for SATA drives.

By default, qpolicy is **on**.

Example:

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

/c*x*/u*x* set storsave=protect|balance|perform [quiet]

This feature only applies to 9000 series SX/SE/SA 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 106.

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.

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.

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

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.
```

/c*x*/u*x* migrate type=*RaidType* [disk=*p:-p*] [group=3|4|5|6|7|8|9|10|11|12|13|14|15|16] [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 **/c*x* add** command. For example, a valid number of disks and parameters must be specified.

The unit to be migrated must be in a normal state (not degraded, initializing, or rebuilding) before starting the migration.

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.

type=*RaidType* specifies the RAID type of the destination unit. Possible unit types include **raid0**, **raid1**, **raid5**, **raid6**, **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 11: Valid Migration Paths

	Destination							
Source	**R0**	**R1**	**R5**	**R6**	**R10**	**R50**	**Single**	**Spare**
R0	Yes	No	Yes	Yes	Yes	Yes	No	No
R1	Yes	No	Yes	Yes	Yes	Yes	Yes	No
R5	Yes	No	Yes	Yes	Yes	Yes	No	No
R6	Yes	No	Yes	Yes	Yes	Yes	No	No
R10	Yes	No	Yes	Yes	Yes	Yes	No	No
R50	Yes	No	Yes	Yes	Yes	Yes	No	No
Single	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Spare	No	No	No	No	No	No	No	No

disk=*p:-p..* consists of a list of ports or VPorts (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 or VPort 0, 1, 2 through 5 (inclusive), 9 and 12.

group=*3|4|5|6|7|8|9|10|11|12|13|14|15|16* 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 through 12. For example **group=3** indicates 3 disks of RAID 5 at the bottom layer of RAID 50.

Note: You can have a maximum of 4 subunits in a RAID 50 unit

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 disks can be grouped into groups of 3, 4, or 6 drives (the group of 4 drives is set by default as it provides the best of net capacity and performance). For 15 disks, the disks can be grouped into groups of 5 drives (3 drive groups would make 5 subunits, you can have a maximum of 4 subunits). For 16 disks, the disks can be grouped into groups of 4 or 8 drives.

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 12: Supported Stripe Sizes

Model	R0	R1	R5	R6	R10	Spare	R50	Single
7/8000	64	N/A	64	N/S	64	N/A	N/S	N/S
	128				128			
	256				256			
	512				512			
	1024				1024			
9000[a]	16	N/A	16	N/S	16	N/A	16	N/S
	64		64	N/S	64		64	
	256		256	N/S	256		256	
9650SE and 9690SA	16	N/A	16		16	N/A	16	N/S
	64		64	64	64		64	
	256		256		256		256	

[a] Models 9500S, 9550SX, and 9590SE

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 (uninterruptible 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 70.

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 67 and “/cx rescan [noscan]” on page 50.

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 should be split into Single units. In this case, u3 is a RAID 1 and the Migrate command splits u3 into u3 and u*x*, each with a RAID type of Single.

Warning: Make sure that no I/O is pending before splitting a mirror. If the RAID 1 is the boot device, you should boot from a different device before splitting the mirror.

Example of capacity expansion

```
//localhost> /c0/u3 migrate type=raid10 disk=10-11 stripe=16
```

Indicates that the destination unit has a RAID type of raid10 and has added the disks 10 and 11 to the disks in the existing unit u3.

Example of migrate output

The following is an example of how migrating units will be displayed. In this example, the report indicates that `/c0/u3` is a migrating unit with 39% completion. The report also indicate that Source Unit `su0` is of type RAID-1 and Destination Unit `du0` is of type RAID-10.

```
3ware CLI> /c0 show

 Unit  UnitType  Status       %RCmpl  %V/I/M  Stripe  Size(GB)  Cache  AVrfy
 ---------------------------------------------------------------------------
 u0    RAID-5    OK            -       -       64K     596.004   ON     OFF
 u2    SPARE     OK            -       -       -       149.042   -      OFF
 u3    Migrator  MIGRATING     -       39      -       149.001   ON     OFF

 VPort   Status           Unit   Size        Blocks        Serial
 ---------------------------------------------------------------
 p0     OK                u0     149.05 GB   312581808     WD-WCANM1771318
 p1     OK                u0     149.05 GB   312581808     WD-WCANM1757592
 p2     OK                u0     149.05 GB   312581808     WD-WCANM1782201
```

```
p3     OK               u0     149.05 GB   312581808     WD-WCANM1753998
p4     OK               u2     149.05 GB   312581808     WD-WCANM1766952
p5     OK               u3     149.05 GB   312581808     WD-WCANM1882472
p6     OK               u0     149.05 GB   312581808     WD-WCANM1883862
p7     OK               u3     149.05 GB   312581808     WD-WCANM1778008
p8     OK               -      149.05 GB   312581808     WD-WCANM1770998
p9     NOT-PRESENT      -      -           -             -
p10    OK               u3     149.05 GB   312581808     WD-WCANM1869003
p11    OK               u3     149.05 GB   312581808     WD-WCANM1762464

3ware CLI> /c0/u0 show

Unit     UnitType  Status        %RCmpl  %V/I/M  VPort  Stripe  Size(GB)
------------------------------------------------------------------------
u3       Migrator  MIGRATING      -       39      -      -       -

su3      RAID-1    OK             -       -       -      -       149.001
su3-0    DISK      OK             -       -       p5     -       149.001
su3-1    DISK      OK             -       -       p7     -       149.001
su3/v0   Volume    -              -       -       -      -       149.001

du3      RAID-10   OK             -       -       -      16K     298.002
du3-0    RAID-1    OK             -       -       -      -       -
du3-0-0  DISK      OK             -       -       p5     -       149.001
du3-0-1  DISK      OK             -       -       p7     -       149.001
du3-1    RAID-1    OK             -       -       -      -       -
du3-1-0  DISK      OK             -       -       p10    -       149.001
du3-1-1  DISK      OK             -       -       p11    -       149.001
du3/v0   Volume    -              -       -       -      -       149.001
```

Port Object Commands

Port Object Messages are commands that provide information and perform actions related to a specific disk, attached to a port (or virtual 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 (9000 series SX/SE/SA only)|
    lspeed (9000 series SX/SE/SA only)|model|
    ncq (9000 series SX/SE/SA only)|serial|smart|status|
    identify (9000 series SX/SE/SA only)|
    driveinfo (9690SA only)
/cx/px show all

/cx/px remove [noscan][quiet]
/cx/px set identify=on|off (9000 series SX/SE/SA only)
```

/c*x*/p*x* show

This command shows summary information about the specified disk attached to port **/c*x*/p*x***.

Example for 9650SE and earlier controllers:

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

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

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

Example for 9690SA controller:

Note that in this case the summary information is in a slightly different format

```
 //localhost> /c3/p1 show

 VPort Status       Unit  Size      Type   Phy Encl-Slot    Model
 --------------------------------------------------------------------------
 p1    OK            u0   149.05 GB   SATA   0 -         WDC WD1600JS-22NCB1a
```

The above output indicates that virtual port (VPort) 1 is associated with a SATA disk that has a status of OK and is part of unit 0. It also indicates that it is associated with controller phy 0, and is directly attached (that is—it is not in an enclosure).

The VPort number provides a handle by which to identify the drive, even though the drive may not be directly attached to the controller.

To find the Block and Serial information, use the specific show attribute command (***/cx/px show capacity*** or ***/cx/px show serial***), or the ***show all*** command.

/c*x*/p*x* 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 (9000 series SX/SE/SA only), lspeed (9000 series SX/SE/SA only), model, ncq (9000 series SX/SE/SA only), serial, smart (SATA only), and status.

/c*x*/p*x* 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 Blocks)
```

/cx/px show driveinfo

This command applies to the 9690SA only. It displays the following information about the drive:

Drive Type can be either SAS or SATA.

Interface Type can be either Direct or Expander. This indicates how the drive is connected to the controller.

Drive Ports indicates the number of physical ports on the drive. Can be 1 or 2, depending on whether the drive is a dual-ported drive. SAS drives can be either single or double ported, while SATA drives have only one port.

Drive Connections indicates the number of connections from the controller to the drive. Can be either 1 or 2.

Additional information is available using the ***/cx/px*** show all command.

Example:

```
//localhost> /c3/p4 show driveinfo
/c3/p4 Drive Type = SAS
/c3/p4 Interface Type = Direct
/c3/p4 Drive Ports = 2
/c3/p4 Drive Connections = 1
```

/c*x*/p*x* 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
```

/c*x*/p*x* show identify

This command only applies to 9000 series SX/SE/SA 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 85.

Example:

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

/c*x*/p*x* show lspeed

This command only applies to 9000 series SX/SE/SA 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
```

/c*x*/p*x* 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
```

/c*x*/p*x* show ncq

This command only applies to 9000 series SX/SE/SA model controllers.

This command displays the NCQ (Native Command Queueing) 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.

Queuing information is not available for SAS drives.

Example:

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

/c*x*/p*x* 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
```

/c*x*/p*x* show smart

SMART data is only applicable to SATA drives. It is not available for SAS drives.

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 00 03 07
 00 9A 96 BC 14 00 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 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.

/c*x*/p*x* 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
```

/c*x*/p*x* show all

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

Example for 9690SA:

```
/c3/p3 show all
/c3/p3 Status = OK
/c3/p3 Model = ST3750640AS
/c3/p3 Firmware Version = 3.AAK
/c3/p3 Serial = 3QD0RGX6
/c3/p3 Capacity = 698.63 GB (1465149168 Blocks)
/c3/p3 WWN = 0000000000000000
/c3/p3 Drive Type = SATA
/c3/p3 Interface Type = Direct
/c3/p3 Drive Ports = 1
/c3/p3 Drive Connections = 1
/c3/p3 Link Speed Supported = 1.5 Gbps
/c3/p3 Link Speed = 1.5 Gbps
/c3/p3 Queuing Supported = Yes
/c3/p3 Queuing Enabled = Yes
/c3/p3 Reallocated Sectors = 0
/c3/p3 Power On Hours = 268
/c3/p3 Temperature = 36~C
/c3/p3 Spindle Speed = 7200 RPM
/c3/p3 Identify Status = N/A
/c3/p3 Belongs to Unit = N/A

/c3/p3 Drive Smart Data:
0A 00 01 0F 00 5E 59 B8 9A 24 0A 00 00 00 03 03
00 5E 5D 00 00 00 00 00 00 00 04 32 00 64 64 51
00 00 00 00 00 00 05 33 00 64 64 00 00 00 00 00
00 00 07 0F 00 4A 3E 88 3F 98 01 00 00 00 09 32
00 64 64 0C 01 00 00 00 00 00 0A 13 00 64 64 00
00 00 00 00 00 00 0C 32 00 64 64 52 00 00 00 00
00 00 BB 32 00 64 64 00 00 00 00 00 00 00 BD 3A
00 64 64 00 00 00 00 00 00 00 BE 22 00 40 38 24
00 1B 28 00 00 00 C2 22 00 24 2C 24 00 00 00 19
00 00 C3 1A 00 3B 38 68 2B 4E 00 00 00 00 C5 12
00 64 64 00 00 00 00 00 00 00 C6 10 00 64 64 00
```

```
00 00 00 00 00 00 C7 3E 00 C8 C8 00 00 00 00 00
00 00 C8 00 00 64 FD 00 00 00 00 00 00 00 CA 32
00 64 FD 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 82 00 AE 01 00 5B
03 00 01 00 01 CA 02 00 00 00 00 00 00 00 00 00
00 00 0B 01 00 00 02 07 07 07 07 06 07 07 07 00
00 00 00 00 00 00 00 01 3A 68 E2 4B 00 00 00 00
00 00 C5 AB 08 58 1D 00 00 00 00 00 00 00 00 00
00 00 00 00 3A 68 E2 4B 00 00 00 00 00 00 00 00
02 00 01 00 00 00 00 00 00 00 C5 60 F4 BC 00 00
00 00 30 00 00 00 00 00 00 00 0B 01 00 00 00 00
C4 55 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 CC
```

/c*x*/p*x* remove [noscan] [quiet]

This command allows you to remove (or export) a port (or drive) **/c*x*/p*x***. 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 bay is available. System hangs and damage can occur if a hot swap bay is not used.

/c*x*/p*x* set identify=on|off

This command applies only to 9000 series SX/SE/SA 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 "/cx/ex/slotx set identify=on|off" on page 95.

Example:

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

Phy Object Commands

These commands are for 9690SA controllers only.

Phy Object Commands are sent to an instance of a controller phy such as ***/c0/phy0***.

This command sets the link speed control of the phy. The possible values are auto, 1.5, and 3.0 for SATA, and auto and 3.0 for SAS. The units are in Gigabits per second (Gbps).

Syntax

```
/cx/phyx set link=auto|1.5|3.0
```

/c*x*/phy*x* set link=auto|1.5|3.0

This command sets the link speed for the specified phy. This will control the speed between the controller and the hard drive or enclosure to which it is connected.

Example:

```
//localhost> /c0/phy0 set link=auto
 Sending link speed control of /c0/phy0 to [auto] ...  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 series 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 series only)

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

/c*x*/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.

/c*x*/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.

/c*x*/bbu show batinst

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

/c*x*/bbu show bootloader

This command shows the BBU's boot loader version.

/c*x*/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 **/c*x*/bbu show lasttest** to check the date of the last test. For information about running a test, see "/cx/bbu test [quiet]" on page 90.

/c*x*/bbu show fw

This command shows the BBU firmware version number.

/c*x*/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 90.

/c*x*/bbu show pcb

This command shows the PCB revision number on the BBU.

/*cx*/bbu show serial

This command shows the BBU serial number.

/*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

/c*x*/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.)

/c*x*/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.)

/c*x*/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>
```

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

/c*x*/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.

/c*x*/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 **/c0/e0** and its elements, such as **/c0/e0/slot0**. Enclosure object elements include slot, fan, and temperature sensor elements.

These commands are supported on 9690SA controllers.

Syntax

```
/cx/ex show
/cx/ex show attribute [attribute ...] where attributes are:
    controllers|diag|slots|fans|temp
/cx/ex show all

/cx/ex/slotx show
/cx/ex/slotx show identify
/cx/ex/slotx set identify=on|off

/cx/ex/fanx show
/cx/ex/fanx/ show identify
/cx/ex/fanx set identify=on|off
/cx/ex/fanx/ set speed=<0|1|2|3|4|5|6|7>

/cx/ex/pwrsx show
/cx/ex/pwrsx/ show identify
/cx/ex/pwrsx set identify=on|off

/cx/ex/tempx show
/cx/ex/tempx/ show identify
/cx/ex/tempx set identify=on|off
```

/cx/ex show

This command is used with the 9690SA controller.

This command shows summary information on the specified enclosure **cx/ex.** This report consists of several parts, depending on the available elements of the enclosures. Typically, the summary consists of the **Enclosure** section listing the set of enclosures, a **Fan** section listing the set of fans, a **Temperature Sensor** section listing of the set of temperature sensor, a **Power Supply** section listing the set of power supplies, and a **Slot** section listing the set of slots.

Example:

```
 //localhost> /c0/e0 show
Encl        Status
 ---------------------------
 /c0/e0      -
Fan         Status          State   Step   RPM     Identify
 ------------------------------------------------------------
 fan0        OK              ON      1      2670    Off
 fan1        OK              ON      1      9500    Off
 fan2        OK              ON      1      8540    Off
 fan3        OK              ON      1      2830    Off
 fan4        OK              ON      1      9120    Off
 fan5        OK              ON      1      8330    Off
 TempSensor    Status         Temperature     Identify
 --------------------------------------------------------
 temp0         OK             41C(105F)       Off
 temp1         OK             38C(100F)       Off
 temp2         OK             34C(93F)        Off
 temp3         OK             38C(100F)       Off
 temp4         OK             38C(100F)       Off
 temp5         OK             34C(93F)        Off
 temp6         NOT-INSTALLED  -               Off
 temp7         NOT-INSTALLED  -               Off
PowerSupply  Status         State  Voltage        Current        Identify
 ---------------------------------------------------------------------------
 pwrs0        OK             on     OK             OK             Off
 pwrs1        OK             on     OK             OK             Off

 Slot      Status          Port          Identify
 --------------------------------------------------
 slot0     OK              /c0/p0        Off
 slot1     NO-DEVICE       -             Off
 slot2     OK              /c0/p1        Off
 slot3     OK              /c0/p2        Off
 slot4     OK              /c0/p3        Off
 slot5     OK              /c0/p4        Off
 slot6     OK              /c0/p5        Off
 slot7     OK              /c0/p6        Off
 slot8     OK              /c0/p7        Off
 slot9     OK              /c0/p8        Off
 slot10    OK              /c0/p9        Off
 slot11    NO-DEVICE       -             Off
```

/cx/e*x* show *attribute* [*attribute* ...]

This command is used with the 9690SA controller.

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, slots, fans, and temp.

/cx/e*x* show controllers

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

Example:

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

/cx/ex/ show diag=helptext

This command is used with the 9690SA controller.

This command allows you to view the enclosure diagnostic log. What is reported by this log will vary depending upon the manufacturer of the enclosure.The enclosure diagnostic log may be requested by 3ware Customer Support to help troubleshoot problems on your controller.

/cx/*ex* show slots

This command is used with the 9690SA controller.

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

Example:

```
 //localhost> c0/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
```

/cx/*ex* show fans

This command is used with the 9690SA controller.

This command lists the fans in enclosure **/cx/ex** and shows their status.

Example:

```
Example: //localhost> /c0/e0 show fans
                                     ---Speed---
 Fan         Status          State   Step   RPM     Identify
 ------------------------------------------------------------
 fan0        OK              ON      1      2670    Off
 fan1        OK              ON      1      9370    Off
 fan2        OK              ON      1      8540    Off
 fan3        OK              ON      1      2810    Off
 fan4        OK              ON      1      9240    Off
 fan5        OK              ON      1      8330    Off
```

/*cx*/*ex* show temp

This command is used with the 9690SA controller.

This command lists the temperature sensors in enclosure **/cx/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> /c0/e0 show temp
TempSensor    Status        Temperature     Identify
 --------------------------------------------------------
 temp0         OK             41C(105F)       Off
 temp1         OK             37C(98F)        Off
 temp2         OK             34C(93F)        Off
 temp3         OK             38C(100F)       Off
 temp4         OK             38C(100F)       Off
 temp5         OK             34C(93F)        Off
 temp6         NOT-INSTALLED  -               Off
 temp7         NOT-INSTALLED  -               Off
```

/cx/e*x* show all

This command is used with the 9690SA controller.

This command shows the current settings of all attributes for enclosure **/cx/e*x***

/cx/e*x*/slot*x* show

This command is used by the 9690SA controller.

This command shows information about the specified **/slot*x*** 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> c0/e0/slot1 show
 Slot      Status          Port          Identify
 ----------------------------------------------------
 slot1     OK              /c0/p1        Yes
```

/cx/e*x*/slot*x* show identify

This command is used by 9690SA controllers.

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

Example:

```
//localhost> c0/e0/slot1 show identify
/e0/slot1 Identify status = on
```

/c*x*/e*x*/slot*x* set identify=on|off

This command is used by 9690SA controllers.

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.

Note: This command is equivalent to "/cx/px set identify=on|off" on page 85.

Example:

```
//localhost> c0/e0/slot1 set identify=on
Sending Identify request to Drive Slot c0/e0/slot0 to [on] ...
Done.
```

/c*x*/e*x*/fan*x* show

This command is used by the 9690SA controller.

This command shows the information about the specified fan element **/fan*x*** in the specified enclosure **c*x*/e*x***. 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> /c0/e0/fan0 show
                                            ---Speed---
 Fan         Status          State   Step   RPM     Identify
 ------------------------------------------------------------
 fan0        OK              ON      1      2700    Off
```

/cx/ex/fan***x*** show identify

This command is used by 9690SA controllers.

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

Example:

```
//localhost> c0/e0/fan1 show identify
c0/e0/fan1 Identify status = on
```

/cx/ex/fan*x* set identify=on|off

This command is used by 9690SA controllers.

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

Setting identify to on will cause the LED associated with that fan to blink.

Example:

```
//localhost> c0/e0/fan1 set identify=on
Sending Identify request to Fan c0/e0/fan1 to [on] ... Done.
```

/c*x*/e*x*/pwrs*x* show

This command is used by the 9690SA controller.

This command shows the information about the specified power supply element **/pwrs*x*** in the specified enclosure **c*x*/e*x***. The power supply name is followed by its status. The possible status values are OK, OVERTEMP-FAILURE, OVERTEMP-WARNING, DC-FAIL. and AC-FAIL. The possible Voltage and Current values are OK, OVER-WARNING, and UNDER-WARNING.

If a power supply is on and no fault has been detected, the status would indicate OK.

/cx/ex/pwrs*x* show identify

This command is used by 9690SA controllers.

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

Example:

```
//localhost> c0/e0/pwrs1 show identify
c0/e0/pwrs1 Identify status = on
```

/cx/ex/pwrs*x* set identify=on|off

This command is used by 9690SA controllers.

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

Setting identify to on will cause the LED associated with that power supply to blink.

Example:

```
//localhost> c0/e0/pwrs1 set identify=on

Sending Identify request to Power Supply c0/e0/pwrs1 to [on] ...
Done.
```

/c*x*/e*x*/temp*x* show

This command is used by the 9690SA controller.

This command shows the information about the specified temperature sensor element **/fan*x*** in the specified enclosure **c*x*/e*x***. The possible status values are OK, OVER-WARNING, OVER-FAIL, UNDER-WARNING, UNDER-FAIL, where OVER denotes over-temperature and UNDER denotes under-temperature. For more information about these statuses, see the *3ware Serial ATA RAID Controller User Guide*

Example:

```
//localhost> /c0/e0/temp0 show
TempSensor    Status         Temperature     Identify
 -------------------------------------------------------
 temp0         OK             42C(107F)       Off
```

/cx/ex/temp*x* show identify

This command is used by 9690SA controllers.

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

Example:

```
//localhost> c0/e0/temp1 show identify
c0/e0/temp1 Identify status = on
```

/cx/ex/temp*x* set identify=on|off

This command is used by 9690SA controllers.

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

Setting identify to on will cause the LED associated with that temp sensor to blink.

Example:

```
//localhost> c0/e0/temp1 set identify=on

Sending Identify request to Temp Sensor c0/e0/temp1 to [on] ...
Done.
```

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. (For an example, see the discussion of the command “help” on page 100.)

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 **/c*x*, /c*x*/u*x*, /c*x*/p*x*, /c*x*/bbu, /e*x*, /e*x*/slot*x*, /e*x*/fan*x***, and **/e*x*/temp*x***), 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 (9000 series
    SX/SE/SA only)|bios|carvesize(9KSX/SE/SA only)|
    ctlbus(9KSX/SE/SA only)|driver|
    drivestatus|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 series)
/cx show verify                           (9000 series)
/cx show selftest                         (9000 series)
/cx show phy                              (9690SA only)
/cx add type=<RaidType> disk=<p:-p..> [stripe=<Stripe>]
    [noscan]
    [nocache][group=<3|4|5|6|7|8|9|10|11|12|13|14|15|16>]
    [autoverify][ignoreECC] [name=string (9000 series)]
    [storsave=<protect|balance|perform[quiet]>(9KSX/SE/SA
    only)] RaidType={raid0, raid1, raid5, raid6
    (9650SE), raid10, raid50, single, spare}
/cx add rebuild=ddd:hh:duration            (9000 series)
/cx add verify=ddd:hh:duration             (9000 series)
/cx add selftest=ddd:hh                    (9000 series)

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

/cx set ondegrade=cacheoff|follow          (9500S only)
/cx set spinup=nn                          (9000 series)
/cx set stagger=nn                         (9000 series)
/cx set autocarve=on|off                   (9000 series)
/cx set carvesize=[1024..2048]          (9KSX/SE/SA only)
/cx set rebuild=enable|disable|<1..5>
                        (enable|disable for 9000 series)
/cx set verify=enable|disable|<1..5>
                        (enable|disable for 9000 series)
/cx set selftest=enable|disable [task=UDMA|SMART](9000
    series)
/cx set autorebuild=on|off             (9KSX/SE/SA only)
/cx set autodetect=on|off disk=<p:-p>|all  (9000 series)
/cx update fw=filename_with_path [force]   (9000 series)
/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]
//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|firmware|memory|model|
    monitor|numdrives|numports|numunits|ctlbus(9000 series
    SX/SE/SA only)|serial|ondegrade (9500S 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) 2004-2006 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
commit  Commit dirty DCB to storage on controller(s).        (Windows only)
/cx       Controller specific commands.
/cx/ux    Unit specific commands.
/cx/px    Port specific commands.
/cx/bbu   BBU specific commands.                (9000 only)
/cx/ex    Enclosure Specific commands.          (9690SA only)

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 24 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 24 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 24 for more information.

help update

This command provides specific update-related help. See “Shell Object Commands” on page 24 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 24 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 24 for more information.

help /c*x*

This command provides specific controller **/c*x*** related help, illustrating various commands associated with the controller **/c*x***. See the section “Controller Object Commands” on page 29 for more information.

help /c*x*/phy*x*

This command provides specific /cx/phyx related help, illustrating various ways to use the /cx/phyx command. See “Phy Object Commands” on page 85.

help /c*x*/u*x*

This command provides specific unit **/c*x*/u*x*** related help, illustrating various commands to use on a unit **/c*x*/u*x***. See the section “Unit Object Commands” on page 61 for more information.

help /c*x*/p*x*

This command provides specific **/c*x*/p*x*** related help, illustrating various ways to use the **/c*x*/p*x*** command. See the section “Port Object Commands” on page 79 for more information.

help /c*x*/bbu

This command provides specific **/c*x*/bbu** related help, illustrating various ways to use the **/c*x*/bbu** command. See the section “BBU Object Commands” on page 86 for more information.

help /c*x/*e*x*

This command provides specific enclosure cx/ex related help, illustrating various commands associated with the enclosure cx/ex. See the section “Enclosure Object Commands” on page 91 for more information.

help /c*x*/e*x*/slot*x*

This command provides specific slot cx/ex/slotx related help, illustrating various ways to use cx/ex/slotx. See the section “Enclosure Object Commands” on page 91 for more information.

help /c*x*/e*x*/fan*x*

This command provides specific fan cx/ex/fanx related help, illustrating various ways to use the cx/ex/fanx command. See the section “Enclosure Object Commands” on page 91 for more information.

help /c*x*/e*x*/temp*x*

This command provides specific temperature sensor cx/ex/tempx related help, illustrating various ways to use the cx/ex/tempx command. See the section “Enclosure Object Commands” on page 91 for more information.

help /c*x*/e*x*/pwrsx

This command provides specific power supply /cx/ex/pwrsx related help, illustrating various ways to use the /cx/ex/pwrsx command. See the Enclosure Element Power Supply section for more.

Command Logging

This feature logs controller commands from both CLI and 3DM2 into a file. You may be asked to supply this logfile to tech support for troubleshooting.

Set the environment variable TW_CLI_LOG to ON or OFF to enable or disable logging of controller commands into a log file called tw_mgmt.log.

By default, TW_CLI_LOG is set to OFF. The command to start command logging varies by operating system.

- For Redhat and SuSE, (bash, ksh, or sh), enter
  ```
  export TW_CLI_LOG=ON
  ```
- For Linux (chs C-shell), enter
  ```
  setenv TW_CLI_LOG ON
  ```
- For Windows, enter
  ```
  set TW_CLI_LOG=O
  ```

In Linux, the log file is in /var/log directory.

In Windows, the log file is in the 3DM2 installation directory if 3DM2 is installed in the system. Otherwise, it is in the current user home directory.

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.

To view the return code for Linux:

At the shell command prompt type:

```
echo $?
```

The screen prints either a 0 or a 1, depending on whether the command was successful or not.

For example, if you had a 3ware controller with an ID of 0, you could type this command:

```
tw_cli /c0 show
(c0 information displayed here)
echo $?
```

```
0
```

If you type:

```
tw_cli /c7 show
error: (CLI003) specified controller does not exist.
echo $?
1
```

This example fails (returns 1) because there is no controller 7.

To view the return code for Windows, in a command window type

```
tw_cli /c0 show
(c0 info displayed here)
if errorlevel 0 echo 0
0

tw_cli /c7 show
error....
if errorlevel 1 echo 1
1
```

This example fails (returns 1) because there is no controller 7.