

man, tw_cli, 3ware, twcli, cli, raid, raidcontroller, wip



tw_cli

NAME

tw_cli(8) – AMCC/3ware Storage Controller Management Command Line Interface.

SYNOPSIS

```
tw_cli           Interactive Mode
tw_cli -f file   Process from a file
tw_cli command  Process single command (batch mode)
```

DESCRIPTION

tw_cli(8) is a Command Line Interface Storage Management Software for AMCC/3ware ATA RAID Controller(s). It provides controller, logical unit and drive management. **tw_cli** can be used in both interactive and batch mode, providing higher-level API (Application Programming Interface) functionalities.

As a way of synchronizing terminologies, logical units (a.k.a. units) are block devices presented to operating systems. A logical unit can be a one-tier, two-tier, or three-tier arrangement. JBOD, 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. A controller has one or many ports (typically 4, 8, 12). Each port is connected to zero or one disk drive.

CLI also supports *comments*. Command lines beginning with **#** denotes start of comment. This feature is mostly useful with batch processing via **-f script** flag.

CLI prompt indicates the current object in focus, expressed in URI (Universal Resource Identifier) syntax consisting of a hostname (**hostname**), **and an object path (/path/path/object) such as elvis/c0/u0**. User can set the focus to a particular object by **focus URI**.

In this version of the CLI release, the *tw_cli* supports a set of primary command syntax (so called new command syntax) and a set of legacy command syntax (so called old command syntax or original command syntax). Note: we reserve the right to discontinue the legacy command syntax in the future releases.

Primary Command Syntax

The primary command syntax will replace the legacy command syntax in the future releases. This new and improved command format follows a general grammar in the form:

Object Message Attributes

Objects are either shell commands or specify a certain *controller, logical unit, port (drive), or battery backup unit (bbu)*. Messages are commands sent to the requested objects and will be a read operation such as *show* or write operation such as *set, delete, add, stop, start, or export*. Attributes specify what values to read or write. Attributes are either **Boolean Attributes** or **Named Attributes**. Value of a **Boolean attribute** is deduced by presence or not. Value of named attributes are expressed in a "key = value" format.

Shell Object Messages

Shell Object Messages are commands (a.k.a. methods/messages) that are sent to the Command Interpreter (a.k.a. Shell/CLI) itself.

show

This command shows a general summary of all detected controllers. Note that the appropriate kernel device drivers should be loaded for the list to show all controllers. The intention is to provide a global view of the environment.

Typical output looks like:

```
//localhost> show
```

Ctl	Model	Ports	Drives	Units	NotOpt	RRate	VRate	BBU
c0	7500-12	12	8	3	1	2	-	-
c1	9506S-12	12	6	1	0	3	5	TESTING

Indicating that Controller 0 is a 7500 model with 12 Ports, with 8 Drives detected (attached), total of 1 Units, with one unit in a NotOpt (Not Optimal) state, RRate(Rebuild Rate) of 2, VRate(Verify Rate) of '-' (Not Applicable), BBU of '-' (Not Applicable). Not Optimal refers to any state except OK and VERIFYING . Other states include INITIALIZING , INIT-PAUSED , REBUILDING , REBUILD-PAUSED , DEGRADED , MIGRATING , MIGRATE-PAUSED , RECOVERY , INOPERABLE , and UNKOWN . show ver This command will show the CLI and API version.

For example:

localhost> show ver CLI Version = 2.00.00.0xx API Version = 2.00.00.0xx localhost> show alarms [reverse] This command shows the alarms or AEN messages of all controllers in the system. The default is to display the most recent message first. The reverse attribute offers the display order to be the most recent message last. show diag This command shows the diagnostic information of all controllers in the system. show rebuild This command displays all rebuild schedules of all the 9000 controllers in the system. show verify This command displays all verify schedules of all the 9000 controllers in the system. show selftest This command displays all self test schedules of all the 9000 controllers in the system. focus Object This command will set the specified object in focus. This command is active in interactive mode only and is provided to reduce typing. Recall that messages (or commands) are sent to objects such as

hostname/c0/u0 show Instead, if the focus is set to hostname/c0/u0, the prompt is changed automatically to reflect this and the user would only have to type show. The concept is similar to being in a particular location in a file system and requesting a listing of the current directory.

object can have the following forms:

hostname/cx/ux specifies the fully qualified URI of an object on host hostname, controller cx, unit ux. hostname specifies root of host hostname. The hostname is the name of the system where your 3ware RAID controllers are. With current releases, the hostname here should be always your system's name.

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

/ specifies the root at the current focused host.

./obj specifies the next level of the object.

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

For example:

localhost> focus elvis.amcc.com elvis.amcc.com> elvis.amcc.com> focus /c0/u0 elvis.amcc.com/c0/u0> elvis.amcc.com/c0/u0> focus .. elvis.amcc.com/c0> elvis.amcc.com/c0> focus ./u0 elvis.amcc.com/c0/u0> elvis.amcc.com/c0> focus / elvis.amcc.com> Note that focus is available as default. You can also set TW_CLI_INPUT_STYLE=OLD in the following to disable the feature. If Bash, then "export TW_CLI_INPUT_STYLE=OLD" If csh, then "setenv TW_CLI_INPUT_STYLE OLD" If Windows, then "set TW_CLI_INPUT_STYLE=OLD" Controller Object Messages Controller Object Messages are commands (a.k.a. methods/messages) that are sent to an instance of a controller such as /c0. /cx show This command shows summary information on the specified controller /cx. This report consists of two to three parts; a Unit summary listing all present units, a Port summary section listing of all present disks and their attached ports, and a BBU summary section listing if a BBU unit is installed on the controller. The Unit summary section lists all present units specifying their Unit Number, Unit type (such RAID 5), Status, usable capacity in Giga (or Tera) Bytes, number of blocks and unit status such as OK , VERIFYING , INITIALIZING , etc. %Cmpl reports percent completion of REBUILDING or VERIFYING , etc., units. Note: If a "*" appears at the end of the status, it means one of its sub-unit is in WARNING state which a recoverable error might be found in the corresponding port. The Port summary section lists all present ports specifying the port number, disk status, unit affiliation, size (in human readable and blocks of 512 bytes), and disk vendor assigned serial number. The BBU summary section lists 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, and its readiness, etc. Additional attributes about controllers, units, ports and disks can be obtained by querying for them explicitly. See other show sub-commands below. Typical output looks like: localhost> /c0 show Unit UnitType Status %Cmpl Stripe Size(GB) Cache AVerify IgnECC

u0 RAID-1 OK - - 149.05 ON OFF OFF u1 RAID-5 OK - 64k 298.22 ON OFF OFF u2 SPARE OK - - 149.05 ON OFF - Port Status Unit Size Blocks Serial

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

Name OnlineState BBUReady Status Volt Temp Hours LastCapTest

bbu On Yes OK OK OK 241 22-Jun-2004 /cx show Attribute Attribute ...

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

/cx show driver

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

Example:

localhost> /c0 show driver /c0 Driver Version = 1.02.00.036 /cx show model This command reports the controller model of controller /cx. Example: localhost> /c0 show model /c0 Model = 7500-12 /cx show firmware

This command reports the firmware version of controller /cx.

Example:

localhost> /c0 show firmware /c0 Firmware Version = FGXX 2.01.00.025 /cx show bios This command reports the BIOS version of controller /cx. Example: localhost> /c0 show bios /c0 BIOS Version = BG9X 2.01.00.026 /cx show monitor

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

Example:

localhost> /c0 show monitor /c0 Monitor Version = BLDR 1.00.00.008 /cx show serial This command reports the serial number of the specified controller /cx. Example: localhost> /c0 show serial /c0 Serial Number = F12705A3240009 /cx show pcb

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

Example:

localhost> /c0 show pcb /c0 PCB Version = Rev3 /cx show pchip This command reports the PCHIP (PCI Interface Chip) version of the specified controller /cx. Example: localhost> /c0 show pchip /c0 PCHIP Version = 1.30-33 /cx show achip

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

Example:

localhost> /c0 show achip /c0 ACHIP Version = 3.20 /cx show numports This command reports the port capacity (number of physical ports) of the specified controller /cx. Example: localhost> /c0 show numports /c0 Number of Ports = 12 /cx show numunits

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

Example:

localhost> /c0 show numunits /c0 Number of Units = 1 /cx show numdrives This command reports the number of drives currently managed by the specified controller /cx. This report does not include (logically) removed/exported drives. Also note that physically removed disk(s) will not be detected unless I/O is performed against the disk. See /cx/px show smart for a workaround. Example: localhost> /c0 show numdrives /c0 Number of Drives = 5 /cx show exportjbod (9000 series)

This command presents the current JBOD Export Policy; "on", "off" or "Not Supported".

Example:

localhost> /c0 show exportjbod /c0 JBOD Export Policy = Not Supported. localhost> /c1 show exportjbod /c1 JBOD Export Policy = on /cx show spinup (9000 series)

This command presents the number of concurrent disks spin up at the power on.

Example:

localhost> /c0 show spinup /c0 Disk Spinup Policy = 1 /cx show ondegrade (9000S only) This command presents 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 force to be off when the unit becomes degraded. Example: localhost> /c0 show ondegrade /c0 Cache on Degraded Policy = Follow Unit Policy /cx show stagger (9000 series)

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

Example:

localhost> /c0 show stagger /c0 Spinup Stagger Time Policy (sec) = 2 /cx show autocarve (9000 series) This command shows the Auto-Carving policy. If the policy is on, all newly created or migrated units larger than carvesize will be automatically carved into multiples of carvesize volumes and 1 remainder volume. Each volume can 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 2 TB filesystems. For 64-bit OS users, there is no need to set the policy to be "on" unless users want to have multiple smaller volumes to the OS . For 32-bit OS users, it is recommended to keep the policy on unless users know their OS supports more than 2 TB disk devices. When autocarve policy is off, all the new unit creation consists of one single volume. Example: localhost> /c0 show autocarve /c0 Auto-Carving Policy = on /cx show carvesize (9000 series)

This command shows the carvesize that Auto-Carving policy needs. The carve size is between 1024 to 2048 GB . Default carvesize is 2048 GB (i.e. 2 TB). See "/cx show autocarve" command above for details.

Example:

localhost> /c0 show carvesize /c0 Auto-Carving Size = 2000 GB /cx show memory This command presents the size of the memory installed on the controller. Example: localhost> /c0 show memory /c0 Memory Installed = 112MB /cx show ctibus (9550SX and 9590SE only)

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

Example:

localhost> /c0 show ctibus /c0 Controller Bus Type = PCI-X /c0 Controller Bus Width = 64 bits /c0 Controller Bus Speed = 133 Mhz /cx show autorebuild (9550SX and 9590SE only) This command shows the Auto-Rebuild policy. If the policy is enabled, the firmware will choose following drives in order to find a candidate for rebuild operation on a degraded unit. 1. Smallest usable capacity spare. 2. Smallest usable unconfigured drive. 3. Smallest usable capacity failed drive. If the policy is disabled, spare drives are the only candidates for an automatic rebuild operation. Example: localhost> /c0 show autorebuild /c0 Auto-Rebuild Policy = on /cx show unitstatus

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

Example:

localhost> /cx show unitstatus Unit UnitType Status %Cmpl Stripe Size(GB) Cache AVerify IgnECC
----- u0 RAID-5 OK - 64K 223.485 OFF OFF ON u1 JBOD OK - - 76.3352 OFF OFF - u2 RAID-0
OK - 64K 148.99 ON ON - /cx show allunitstatus This command presents a count of Total and Not Optimal units managed by the specified controller /cx. See "Shell Object Messages" for more on Not Optimal definition. Example: localhost> /c0 show allunitstatus

/c0 Total Optimal Units = 2 /c0 Not Optimal Units = 0 /cx show drivestatus

This command presents a list of drives, port assignment, vendor signature, size, status, and unit membership/affiliation.

Example:

localhost> /c0 show drivestatus Port Status Unit Size Blocks Serial ----- p0 OK u0 149.05 GB
312581808 3JS0TF14 p1 OK u0 149.05 GB 312581808 3JS0TETZ p2 OK u1 149.05 GB 312581808 3JS0VG85 p3 OK u1 149.05 GB 312581808
3JS0VGCY p4 OK u1 149.05 GB 312581808 3JS0VGGQ p5 OK u2 149.05 GB 312581808 3JS0VH1P p6 OK - 149.05 GB 312581808 3JS0TF0P
p7 OK - 149.05 GB 312581808 3JS0VF43 p8 OK - 149.05 GB 312581808 3JS0VG8D p9 NOT-PRESENT - - - - p10 NOT-PRESENT - - - - p11 NOT-
PRESENT - - - - /cx show all This command shows the current setting of all attributes. /cx add type=<RaidType> disk=<p:-p>
[stripe=Stripe] [noscan] [group=<3 | 4 | 5 | 6 | 7 | 8>] [nocache] [autoverify] [ignoreECC] [name=string]
[storsave=<protect | balance | perform>] This command allows you to add a new unit or create a unit on the specified controller /cx, of type
RaidType, optional stripe size of Stripe, using one or many disks specified by disk=p:-p. By default the host operating system will be
informed of the new block device and write cache 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 group=3 | 4 | 5 | 6 | 7 | 8 attribute. Upon the success of the new unit creation, a unique serial number
is also assigned to the new unit. Please refer to commands /cx/ux show serial for checking. Note: The default of the unit creation sets write
cache to "on" for performance reason. However, if there is no BBU available for the controller, a warning is sent to standard error. Since this
command is by far the richest command, it deserves more details. /cx is the controller name as in /c0, /c1, etc. type=RaidType consists of
logical unit type as in raid0, raid1, raid5, raid10, raid50, single, spare, and JBOD (7000/8000 only). For example type=raid50. The following
table illustrates supported types and controller models. Model | Raid0 | Raid1 | Raid5 | Raid10 | JBOD | Spare | Raid50 | Single |
---+---+---+---+---+---+---+---+---+---+ 7K/8K | Y | Y | Y | Y | Y | Y | N | N | ---+---+---+---+---+---+---+---+---+---+---+
+---+---+---+ 9K | Y | Y | Y | Y | N | Y | Y | Y | ---+---+---+---+---+---+---+---+---+---+---+ 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 ":" or "-" as port index separators. A dash indicates a range and can be mixed with ":". For example disk=0:1:2-5:9:12 indicates port
0, 1, 2 thru 5 (inclusive), 9 and 12. stripe=Stripe consists of the stripe size to be used. The following table illustrates the supported and
applicable stripes on unit types and controller models. Stripe size of units are in K (kilo bytes). Model | Raid0 | Raid1 | Raid5 | Raid10 |
JBOD | Spare | Raid50 | Single | ---+---+---+---+---+---+---+---+---+---+ 7K/8K | 64 | N/A | 64 | 64 | N/A | N/A |
N/S | N/S | | 128 | | 128 | | | | 256 | | 256 | | | | 512 | | 512 | | | | 1024 | | 1024 | | | |
---+---+---+---+---+---+---+---+---+---+ 9K | 16 | N/A | 16 | 16 | N/A | N/A | 16 | N/A | | 64 | | 64 | 64 | | 64 |
| 256 | | 256 | 256 | | 256 | | ---+---+---+---+---+---+---+---+---+---+---+ group=3 | 4 | 5 | 6 | 7 | 8 consists of the
number of disks per group for a Raid 50 type. Note, this attribute can only be used when type=raid50. Recall that a RAID-50 is a multi-tier
array. At the most bottom layer, N number of disks per group are used to form the RAID-5 layer. These RAID-5 arrays are then integrated
into a RAID-0. This attribute allows you to specify the number of disks in the RAID-5 level. Valid values are 3, 4, 5, 6, 7 and 8. 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 . However given 12 disks, specifying 3 will create four RAID-5 under the RAID-0 level. Given 6 disks and grouping of 6 is not allowed, as you'll basically be creating a RAID-5 . The default group varies based on number of disks. For 6 & 9 disks, default is group=3. For 8 disks, default is group=4. For 10 or 15 disks, default is group=5. For 12 or 16 disks, default is group=4. For 14 disks, default is group=7. Case of 12 disks could be grouped with group=3, group=4, or group=6. Group=4 was set by default as it provides best net capacity and performance. Case of 15 disks could be grouped with group=3 or group=5. And case of 16 disks could be grouped with group=4 and group=8. noscan attribute instructs CLI not to notify OS of creation of the new unit. By default CLI will inform the OS . One application of this feature is to avoid the OS from creating block special devices such as /dev/sdb and /dev/sdc as some implementations might create naming fragmentation and creating a moving target. nocache attribute instructs CLI disable the write cache on the newly created unit. Enabling write cache increases performance at the cost of high-availability. No cache is recommended when no BBU or UPS is installed. autoverify attribute enables the autoverify attribute on the unit that is to be created. For more details on this feature, refer to "cx/ux set Commands" section of this document. This feature is not supported on model 7000/8000. On model 9000, JBOD autoverify attribute is not persistent (does not survive reboots). ignoreECC attribute enables the ignoreECC/OverwriteECC attribute on the unit that is to be created. For more details on this feature, refer to "cx/ux set Commands" section of this document. The following table illustrates the supported Model-Unit Type. This table only applies to setting this feature at unit creation time. Generally ignoreECC applies to redundant units. Model | Raid0 | Raid1 | Raid5 | Raid10 | JBOD | Spare | Raid50 | Single | ---+---+---+---+---+---+---+---+---+---+ 7K/8K | N | N | N | N | N | N | N | N | N | ---+---+---+---+---+---+---+---+---+---+ 9K | N | Y | Y | Y | N | N | Y | N | ---+---+---+---+---+---+---+---+---+---+ name=string attribute allows user to name the new unit. The maximum characters allowed for the string are 21. No space is allowed within the string. If user likes to use some special characters which the OS command shell reserves such as '<', '>', '!', and '&', etc in the name string, the user has to use quote "" around the name string in order to bypass the command shell. User can change the name of the unit any time after the unit creation. This is a feature for 9000 or above series of controllers. Please refer to commands /cx/ux set name=string for changing the name and /cx/ux show name for checking. storsave=protect | balance | perform attribute allows user to set the storsave policy of the new unit. It is a 9550SX and 9590SE only feature. Please refer to command /cx/ux set storsave=protect | balance | perform for detail. /cx rescan [noscan] This command instructs the controller to rescan all ports and reconstitute all units. The controller will update its list of ports (attached disks), and visits 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 OS of the unit discovery. Default is to inform the OS . 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]. Note: Does not import non-JBOD on 7000/8000 models. /cx commit

This command instructs the controller to commit its dirty DCBs to persistent storage (ie disks). While controller is processing I/O requests against underlying disks, an in-transaction bit is set. If a failure (such as power failure) is experienced, subsequent read from the disks, will inform the controller that an un-clean shutdown took place. This command allows the end user to complete all pending I/Os on disks and clear the in-transaction bit.

Typical application of this feature is when an application is using a given unit in raw mode (such as databases) and user would like to shutdown the host (Including UPS post failure automations). This command can then expedite the process by instructing the controller to finish pending requests, clear DCB 's in-transaction flag as we are going down.

Note that block devices (cooked devices) do not require this and clients of block devices (such as file systems) will send its own shutdown request to the devices.

This command only applies to Windows operating system. /cx flush

This command allows you to flush the write cache on all units associated with the /cx controller

/cx show alarms [reverse]

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

Alarms generated on 7000/8000 models do not have dates, as such you'll see a '-' (read not-applicable) in "Date" column. Also on 7000/8000 models, the alarm message, contain the severity as well, hence the "Severity" column is showing a '-' as well.

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

Typical output looks like:

```
localhost> /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 /cx show diag This command extracts controller diagnostics suitable for technical support usage. Note that some characters might not be printable or rendered correctly (human readable). It is recommended to save this output to a file, where it can be communicated to tech support or further studied with Linux utilities like od(1). Example: $ tw_cli /c0 show diag > diag.txt /cx show rebuild Model 9000 series support background tasks such as rebuild, verify, or self test activities. For each activity, up to 7 tasks can be registered,
```

known as slots 1 through 7. Each task activity can be managed by a set of commands including add, del, show and set. Background tasks have a slot id, start day, hour, duration, and status attributes. Rebuild activity attempts to (re)synchronize all members of redundant units such as RAID-1, RAID-10, RAID-5 and RAID-50. Rebuilds can be started manually or automatically if a spare has been defined. Scheduled rebuilds will take place during the scheduled window, if enabled. This command displays the current rebuild background task schedule as illustrated below. \$ tw_cli /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 ---- 4 ---- 5 ---- 6 Mon 1:00am 4
hr(s) disabled 7 Sun 12:00am 1 hr(s) disabled Status "disabled" indicates that the controller will not use the tabled schedules. Note: The
rebuild schedules are also applicable to initialization and migration processes. For example: If a unit is in the initialization state at noon on
Wed, the tabled schedule above is show in the following: $ tw_cli /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 ---- 4 ---- 5 ---- 6 Mon 1:00am 4
hr(s) disabled 7 Sun 12:00am 1 hr(s) disabled $ tw_cli /c1 show Unit UnitType Status %Cmpl Stripe Size(GB) Cache AVerify IgnECC
----- u0 RAID-5 INITIALIZING 0 64K 521.466 ON OFF OFF Port Status Unit Size Blocks Serial
----- p0 NOT-PRESENT - - - p1 OK u0 76.33 GB 160086528 Y2NXL7FE p2 NOT-PRESENT - - - p3 OK u0
76.33 GB 160086528 Y2NXLB9E p4 NOT-PRESENT - - - p5 OK u0 76.33 GB 160086528 Y2NXQPZE p6 NOT-PRESENT - - - p7 OK u0 76.33
GB 160086528 Y2NXM4VE p8 OK u0 74.53 GB 156301488 3JV3WTSE p9 OK u0 74.53 GB 156301488 3JV3WRHC p10 OK u0 74.53 GB
156301488 3JV3WQLQ p11 OK u0 74.53 GB 156301488 3JV3WQLZ Name OnlineState BBUReady Status Volt Temp Hours LastCapTest
----- bbu On Yes OK OK OK 0 xx-xxx-xxxx Then if the user enables the tabled schedules, the
unit initialization will be paused until next scheduled slot comes. $ tw_cli /c1 set rebuild=enable Enabling scheduled rebuilds on controller
/c1 ...Done. $ tw_cli /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 %Cmpl Stripe Size(GB) Cache AVerify IgnECC
----- u0 RAID-5 INIT-PAUSED 0 64K 521.466 ON OFF OFF Port Status Unit Size Blocks Serial
----- p0 NOT-PRESENT - - - p1 OK u0 76.33 GB 160086528 Y2NXL7FE p2 NOT-PRESENT - - - p3 OK u0
76.33 GB 160086528 Y2NXLB9E p4 NOT-PRESENT - - - p5 OK u0 76.33 GB 160086528 Y2NXQPZE p6 NOT-PRESENT - - - p7 OK u0 76.33
GB 160086528 Y2NXM4VE p8 OK u0 74.53 GB 156301488 3JV3WTSE p9 OK u0 74.53 GB 156301488 3JV3WRHC p10 OK u0 74.53 GB
156301488 3JV3WQLQ p11 OK u0 74.53 GB 156301488 3JV3WQLZ Name OnlineState BBUReady Status Volt Temp Hours LastCapTest
----- bbu On Yes OK OK OK 0 xx-xxx-xxxx /cx show verify Model 9000 series support
```

background tasks such as rebuild, verify, or self test activities. For each activity, up to 7 tasks can be registered, known as slots 1 through 7. Each activity can be managed by a set of commands including add, del, show and set a task. Background tasks have a slot id, start-day-time, duration, status attributes. Verify activity attempts to verify all units based on their unit type. Verifying RAID-1 involves checking that both drives contain the exact data. On RAID-5, the parity information is used to verify data integrity. RAID-10 and 50 are composite types and follow their respective array types. On the 9000 series, non-redundant units such as RAID-0, JBOD, single, and spare, are also verified (by reading and reporting un-readable sectors). This command displays the current verify background task schedule as illustrated below. \$ tw_cli /c1 show verify Verify Schedule for Controller /c1

```
===== Slot Day Hour Duration Status
----- 1 Mon 2:00am 4 hr(s) disabled 2 - - - 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 Status "disabled" indicates that
the controller will not use the tabled schedules. /cx show selftest Model 9000 series support background tasks such as rebuild, verify, or self
test activities. For each activity, up to 7 tasks can be registered, known as slots 1 through 7. Each activity can be managed by a set of
commands including add, del, show and set a task. Background tasks have a slot id, start-day-time, duration, status attributes. selftest
activity provides two types of selftests; UDMA (Ultra Direct Memory Access) and SMART (Self Monitoring Analysis and Reporting). Both self
tests are checked once each day by default. UDMA self test entails checking the current parallel ATA bus speed (between controller and
attached disk), which could have been throttled down during previous operations, and increases the speed for best performance (usually
one level higher). Possible speeds include 33, 66, 100 and 133 Mhz. Note that the UDMA selftest is not applicable (or required) with SATA
drives, but is left enabled by default. SMART selftest instructs the controller to check certain SMART supported thresholds by the disk
vendor. An AEN is logged to the alarms page if a drive reports a SMART failure. The failing drive should be replaced if this error occurs. This
command displays the current selftest background task schedule as illustrated below. $ tw_cli /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 /cx add
rebuild=ddd:hh:duration This command adds a new background rebuild task to be executed on the day ddd (where ddd is Sun, Mon, Tue,
Wed, Thu, Fri, and Sat), at the hour hh (range 0 .. 23), for a duration of duration (range 1 .. 24) hours. This command will fail if no (empty)
slot is available. For "rebuild" background task description, see command /cx show rebuild. For example: $ tw_cli /c1 add rebuild=Sun:16:3
Will add a rebuild background task schedule to be executed on Sundays at 4:00 PM for a duration of 3 hours. /cx add verify=ddd:hh:duration
This command adds a new background verify task to be executed on 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. This command will fail if no (empty) slot is available. For "verify"
background task description, see command /cx show verify. For example: $ tw_cli /c1 add verify=Sun:16:3 Will add a verify background task
schedule to be executed on Sundays at 4:00 PM for a duration of 3 hours. /cx add selftest=ddd:hh This command adds a new background
selftest task to be executed on 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. This command will fail if no (empty) slot is available. For "selftest"
background task description, see command /cx show selftest. For example: $ tw_cli /c1 add selftest=Sun:16 Will add a selftest background
task schedule to be executed on Sundays at 4:00 PM . /cx del rebuild=slot_id This command will remove (or unregister) the rebuild
background task in slot slot_id. For "rebuild" background task description, see command /cx show rebuild. For example: $ tw_cli /c1 del
rebuild=2 Will remove rebuild background task in slot 2. WARNING: If all timeslots are removed, be sure to also disable the schedule.
Otherwise the applicable background task will never occur. /cx del verify=slot_id This command will remove (or unregister) the verify
background task in slot slot_id. For "verify" background task description, see command /cx show verify. For example: $ tw_cli /c1 del
verify=3 Will remove rebuild background task in slot 3. WARNING: If all timeslots are removed, be sure to also disable the schedule.
```

Otherwise the applicable background task will never occur. `/cx del selftest=slot_id` This command will remove (or unregister) the selftest background task in slot `slot_id`. For "selftest" background task description, see command `/cx show selftest`. For example: `$ tw_cli /c1 del selftest=3` Will remove selftest background task in slot 3. **WARNING:** If all timeslots are removed, be sure to also disable the schedule. Otherwise the applicable background task will never occur. `/cx set rebuild=enable | disable | 1..5` This command will enable or disable all rebuild background tasks on controller `/cx`. When enabled, only tabled scheduled tasks will be followed (or used). Any previous on-demand background tasks will be ignored. This command also allows you to set priority of rebuild vs I/O operations. Setting this value to 1 implies that rebuilds should consume more resources (cpu time, I/O bandwidth) to complete its task. Conversely setting this value to 5 implies that I/O has higher priority and rebuild. This command applies to 7000, 8000, and 9000 models. For 7/8000 series, the rebuild rate also applies to verify and mediascan tasks. For "rebuild" background task description, see command `/cx show rebuild`. `/cx set verify=enable | disable | 1..5` This command will enable or disable all verify background tasks on controller `/cx`. When enabled, only tabled scheduled tasks will be followed (or used). Any previous on-demand background tasks will be ignored. This command allows you to set priority of verification vs I/O operations. Setting this value to 1 implies fastest verify, and 5 implies fastest I/O. Note that this feature only applies to 9000 models. For "verify" background task description, see command `/cx show verify`. `/cx set selftest=enable | disable [task=UDMA | SMART]` This command will enable or disable all or a particular task=`selftest_task` (UDMA or SMART) on a specified controller `/cx`. When enabled, only specified task=`selftest_task` task will be performed during a scheduled timeslot. If no task is specified, the command is applicable to both tasks. For "selftest" background task description, see command `/cx show selftest`. For example: `$ tw_cli /c0 selftest=enable task=UDMA` Will enable UDMA selftest on controller `c0`. `/cx set exportjbod=on | off` This command allows you to set the JBOD Export Policy to on or off. If JBOD export policy is off, CLI would not be able to create JBODs and during reboot, the firmware will not export JBOD units to the operating system. JBOD Export Policy is only supported on 9000 series controllers. Previous models did not have such a policy enforcement feature. `/cx set ondegrade=cacheoff | follow (9000S only)` This command allows you to set a controller based cache policy. If the policy is set to `cacheoff`, then if a unit is degraded, firmware will disable the write-cache on the degraded unit, regardless of what the unit-based policy is. If the policy is set to `follow`, then if a unit is degraded, firmware will follow whatever policy has been set for that unit. `/cx set spinup=nn` This command allows you to set a controller based disk spin up policy. The value must be a positive integer between 1 to the number of disks/ports supported on the controller (e.g. 4, 8, 12, 16). This policy is used to stagger spin ups of disks at boot time in order to spread the power consumption on the power supply. For example, given a spin up policy of 2, the controller will spin up two disks at a time, pause, and then spin up another 2 disks, etc. The amount of time to pause can be specified with the spin up stagger time policy. `/cx set stagger=nn` This command allows you to set a controller based disk spin up stagger time policy. The value must be a positive integer between 0 to 60 seconds. This policy in conjunction with disk spin up policy specifies how the controller should spin up disks at boot time. `/cx set autocarve=on | off (9000 series)` This command allows you to set the Auto-Carving policy to be on or off. When the Auto-Carving policy is on, any unit larger than the `carvesize` is created or migrated into one or more `carvesize` volumes and a remaining volume. Each volume can 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 2 TB filesystems. For example a 3 TB array would be configured into a 2 TB and a 1 TB volumes with default `carvesize`. For a 5 TB array, two 2 TB volumes would be created plus a 1 TB volume. When `autocarve` policy is off, all the new unit creation or migration consists of one single volume. Example: `localhost> /c0 set autocarve=on` Setting Auto-Carving Policy on /c0 to on ... Done. `/cx set carvesize=[1024..2048]` (9000 series)

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

Example:

```
localhost> /c0 set carvesize=2000 Setting Auto-Carving Size on /c0 to 2000 GB ... Done. /cx set autorebuild=on | off (9550SX and 9590SE only) This command sets Auto-Rebuild policy to be on or off. If the policy is on, the firmware will choose drives in the following priority order for a candidate of a rebuild operation (on a degraded unit). 1. Smallest usable capacity spare. 2. Smallest usable unconfigured drive. 3. Smallest usable capacity failed drive. If the policy is off, spares are the only candidates for rebuild operations. Example: localhost> /c0 set autorebuild=on Setting Auto-Rebuild Policy on /c0 to on ... Done. /cx start mediascan
```

This command applies to 7000/8000 controllers. It provides media scrubbing for validating functionality of a disk. This includes bad block detection and remapping, etc. The commands starts a media scan operation on the specified controller `/cx`.

```
/cx stop mediascan
```

This command applies to 7000/8000 controllers. It provides media scrubbing for validating functionality of a disk. This includes bad block detection and remapping, etc. The commands stops a media scan operation on the specified controller `/cx`.

Logical Disk Object Messages

Logical Disk Object Messages are commands (a.k.a. methods/messages) that are sent to an instance of a Logical Disk (a.k.a. unit) such as `/c0/u0`. `/cx/ux show`

This command shows summary information on the specified unit `/cx/ux`. If the unit consists of sub-units as with the case of RAID-10, and RAID-50 arrays, 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 (default is 2 TB - 1), multiple volumes will be created and will be displayed at the end of the summary information.

One application of this 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.

Example:

```
localhost> /c0/u0 show Unit UnitType Status %Cmpl Port Stripe Size(GB) Blocks ----- u0 RAID-50
OK - - 64K 596.05 1249921024 u0-0 RAID-5 OK - - 64K - - u0-0-0 DISK OK - p0 - 149.10 312481280 u0-0-1 DISK OK - p2 - 149.10 312481280
u0-0-2 DISK OK - p3 - 149.10 312481280 u0-1 RAID-5 OK - - 64K - - u0-1-0 DISK OK - p4 - 149.10 312481280 u0-1-1 DISK OK - p5 - 149.10
```

```
312481280 u0-1-2 DISK OK - p6 - 149.10 312481280 localhost> /c0/u1 show
```

```
Unit UnitType Status %Cmpl Port Stripe Size(GB) Blocks
```

```
u1 RAID-0 OK - - 64K 3576.06 7499550720 u1-0 DISK OK - p0 - 298.01 624962560 u1-1 DISK OK - p1 - 298.01 624962560 u1-2 DISK OK - p2 - 298.01 624962560 u1-3 DISK OK - p3 - 298.01 624962560 u1-4 DISK OK - p4 - 298.01 624962560 u1-5 DISK OK - p5 - 298.01 624962560 u1-6 DISK OK - p6 - 298.01 624962560 u1-7 DISK OK - p7 - 298.01 624962560 u1-8 DISK OK - p8 - 298.01 624962560 u1-9 DISK OK - p9 - 298.01 624962560 u1-10 DISK OK - p10 - 298.01 624962560 u1-11 DISK OK - p11 - 298.01 624962560 u1/v0 Volume - - - - 2047.00 4292870144 u1/v1 Volume - - - - 1529.06 3206680576 /cx/ux show Attribute Attribute ...
```

This command shows the current setting of the given attribute(s). One or many attributes can be requested. An invalid attribute will terminate the loop. Possible attributes are: `initializestatus`, `name(9000 series)`, `qpolicy(9550SX and 9590SE Only)`, `rebuildstatus`, `serial(9000 series)`, `status`, `storsave(9550SX and 9590SE Only)`, `verifystatus`, `volumes(9000 series)`. The attributes `volumes`, `name`, and `serial` are applied to 9000 series controllers; and the attributes `qpolicy` and `storsave` are only applied to 9550SX and 9590SE controllers.

```
/cx/ux show status
```

This command presents the status of the specified unit.

Example:

```
localhost> /c0/u0 show status /c0/u5 status = OK /cx/ux show rebuildstatus This command presents the rebuildstatus (if any) of the specified unit. Example: localhost> /c0/u5 show rebuildstatus /c0/u5 is not rebuilding, its current state is OK /cx/ux show verifystatus
```

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

Example:

```
localhost> /c0/u5 show verifystatus /c0/u5 is not verifying, its current state is OK /cx/ux show initializestatus This command presents the initializestatus (if any) of the specified unit. Example: localhost> /c0/u5 show initializestatus /c0/u5 is not initializing, its current state is OK /cx/ux show volumes (9000 series)
```

This command presents the number of volumes of the specified unit.

Example:

```
localhost> /c0/u5 show volumes /c0/u5 Volume(s) = 2 /cx/ux show name (9000 series) This command presents the name (if any) of the specified unit. Example: localhost> /c0/u5 show name /c0/u5 Name = Joe /cx/ux show serial (9000 series)
```

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

Example:

```
localhost> /c0/u5 show serial /c0/u5 Serial Number = 12345678901234567890 /cx/ux show qpolicy (9550SX and 9590SE only) This command presents the queue policy of the firmware. If the queue policy is on, the firmware utilizes the drive queueing policy. If some of the drives do not support any queueing policy, this policy will have no effect on those drives. Example: localhost> /c0/u5 show qpolicy /c0/u5 Command Queuing Policy = on /cx/ux show storsave (9550SX and 9590SE only)
```

This command presents the `storsave` policy (`protect` | `balance` | `perform`) of the firmware on the unit.

For detail, see `/cx/ux set storsave=protect` | `balance` | `perform` (9550SX and 9590SE only).

Example:

```
localhost> /c0/u5 show storsave /c0/u5 Command Storsave Policy = protect /cx/ux show all This command shows the current setting of all above attributes. If the Auto-Carving policy was on at the time the unit was created and the unit is over the carve size (default is 2 TB - 1), multiple volumes will be created and will be displayed at the end of the summary information. Example: localhost> /c0/u1 show all /c0/u1 status = OK /c0/u1 is not rebuilding, its current state is OK /c0/u1 is not verifying, its current state is OK /c0/u1 is not initializing, its current state is OK /c0/u1 volume(s) = 2 /c0/u1 name = 1234567 /c0/u1 serial number = C6CPR7JMF98DA8001DF0
```

```
Unit UnitType Status %Cmpl Port Stripe Size(GB) Blocks
```

```
u1 RAID-0 OK - - 64K 3576.06 7499550720 u1-0 DISK OK - p0 - 298.01 624962560 u1-1 DISK OK - p1 - 298.01 624962560 u1-2 DISK OK - p2 - 298.01 624962560 u1-3 DISK OK - p3 - 298.01 624962560 u1-4 DISK OK - p4 - 298.01 624962560 u1-5 DISK OK - p5 - 298.01 624962560 u1-6 DISK OK - p6 - 298.01 624962560 u1-7 DISK OK - p7 - 298.01 624962560 u1-8 DISK OK - p8 - 298.01 624962560 u1-9 DISK OK - p9 - 298.01 624962560 u1-10 DISK OK - p10 - 298.01 624962560 u1-11 DISK OK - p11 - 298.01 624962560 u1/v0 Volume - - - - 2047.00 4292870144 u1/v1 Volume - - - - 1529.06 3206680576 /cx/ux export [noscan] [quiet]
```

This command allows you to export (or remove) a unit. Exporting a unit will instruct the firmware to remove the specified unit from its pool of managed units, but retains the DCB (Disk Configuration Block) meta-data. As such the unit can later be imported back. `noscan` is used to

not inform the OS of this change. Default is to inform the OS . The quiet option is for non-interactive mode.

Use caution when using this command. Units that are currently in use or mounted cannot be removed. /cx/ux del [noscan] [quiet]

This command allows you to delete a unit. Deleting a unit not only remove the specified unit from the controller’s list of managed units, but also destroys the DCB (Disk Configuration Block) meta–data. Ports (or disks) associated with this unit will now be part of the free pool of managed disks. In another words, once the unit is deleted, all the data on the unit can not be recovered. noscan is used to not inform the OS of this change. Default is to inform the OS . The quiet option is for non-interactive mode.

Use caution when using this command. This is a destructive command and should be used with extreme care. Units that are currently in use or mounted should not be deleted. /cx/ux start rebuild disk=p [ignoreECC]

This command allows you to rebuild a DEGRADED unit by using the specified disk=p. Rebuild only applies to redundant arrays such as RAID–1 , RAID–5 , RAID–10 and RAID–50 . During rebuild, bad sectors on the source disk will cause the rebuild to fail. You can allow for the operation to continue via ignoreECC. Rebuild process is a background task and will change the state of a unit to REBUILDING . Various show commands also show a percent completion as rebuilding progresses.

Note that the disk to be used to rebuild a unit, must be a SPARE or unconfigured disk. /cx/ux start verify

This command starts a background verification process on the specified unit /cx/ux. The following shows the supported matrix 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 that controller model.

Model	Raid0	Raid1	Raid5	Raid10	Raid50	Single	JBOD	Spare	9K
7K/8K	No	Yes	Yes	Yes	N/A	N/A	No	No	Yes
	Yes	Yes	Yes	Yes	N/A	N/A	No	No	Yes

Note that if subsequent to this command, one enables the background verify task to follow the scheduled slots, then this on-demand task will be paused until the next scheduled timeslot. /cx/ux pause rebuild

This command allows you to pause the rebuild operation on the specified REBUILDING unit /cx/ux. This feature is intended for model 7000 and 8000 only. Model 9000 has an on-board scheduler where rebuild operations can be scheduled to take place at specified start and stop times.

Rebuild pause function is provided to enable 7000/8000 users to achieve functionality with use of OS provided schedulers such as cron(8) or, at(1) in Linux or user supplied programs. /cx/ux resume rebuild

This command allows you to resume the rebuild operation on the specified unit /cx/ux. This feature is intended for model 7000 and 8000 only. Model 9000 has an on-board scheduler where rebuild operations can be scheduled to take place at specified start and stop times.

Rebuild resume function is provided to enable 7000/8000 users to achieve similar functionality with use of OS provided schedulers such as cron(8) or, at(1) in Linux or user supplied programs. /cx/ux stop verify

This command stops a background verification process on the specified unit /cx/ux. The following shows the supported matrix 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 that controller model.

Model	Raid0	Raid1	Raid5	Raid10	Raid50	Single	JBOD	Spare	9K
7K/8K	No	Yes	Yes	Yes	N/A	N/A	No	No	Yes
	Yes	Yes	Yes	Yes	N/A	N/A	No	No	Yes

Note that if subsequent to this command, one enables the background verify task to follow the scheduled slots, then this on-demand task will be paused until the next scheduled timeslot. /cx/ux flush

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

/cx/ux set autoverify=on | off

This command allows you to turn on/off the autoverify operation on a specified unit /cx/ux. Once the autoverify=on, the RAID firmware will pick a time to start the verify process on the unit. If the allocated schedule windows is enabled, the verify process becomes active during the scheduled windows. Otherwise, the firmware will decide when the verify needs to be paused or restarted again before it completes.

You can use the show verify command to display the existing schedule windows. The autoverify operation is a continuous verify operation, which takes place within the existing schedule windows (displayed with /cx show verify) if the schedule is enabled. While the “/cx show verify” command allows you to see the time for the verify operation, this command allows you to enable or disable the autoverify operation on the specified unit. This feature only applies to 9000 models. /cx/ux set cache=on | off [quiet]

This command allows you to turn the write cache on a specified unit /cx/ux, on or off. This feature is supported on both 7000/8000 and 9000 models. The quiet option is for non-interactive mode. It can be used in conjunction with set cache=on command when the controller has no BBU installed. The following is the Raid Type-Model support matrix.

Model	Raid0	Raid1	Raid5	Raid10	Raid50	Single	JBOD	Spare	9K
7K/8K	Yes	Yes	Yes	Yes	N/A	N/A	Yes	No	Yes
	Yes	Yes	Yes	Yes	No	No	Yes	No	Yes

/cx/ux set ignoreECC=on | off

This command allows you to set the ignoreECC policy for a given unit such that during rebuild of the specified unit, which could begin automatically (if the unit is degraded and spare has been defined) or manually, to be applied to the rebuild operation. Setting overwriteECC to on means ignoreECC. This feature only applies to 9000 models.

```
/cx/ux set name=string (9000 series)
```

This command allows you to name the unit to an arbitrary name upto 21 characters. No space is allowed within the string. If user likes to use some special characters which the OS command shell reserves such as '<', '>', '!', and '&', etc in the name string, the user has to use quote "" around the name string in order to bypass the command shell. Users can use this name in conjunction with the unit serial number (which created at the unit creation time) to cross reference with the unit. It is user's responsibility to give unique or redundant names on all units. This feature only applies to 9000 models.

Note: Users can also name a JBOD . But the naming information will not persist across power cycles or resets due to the nature of JBODs. /cx/ux set qpolicy=on | off (9550SX and 9590SE only)

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

Example:

```
localhost> /c0/u5 set qpolicy = on Setting Command Queuing Policy for unit /c0/u5 to [on] ... Done. /cx/ux set storsave=protect | balance | perform [quiet] (9550SX and 9590SE only) This command sets the storsave policy of the firmware to be either protect, balance, or perform when the unit write cache is enabled. This feature is available only within 9550SX and 9590SE controller family or above. There is a tradeoff among the available settings. The following description about the settings should help you to decide which one is suitable to you and your application. The protect mode is the default setting. protect - provide the maximum data protection among the controller settings. When user sets storsave to protect mode, it means : 1. "Write Cache" will be disabled when the unit becomes " DEGRADED ", 2. all data flushing from controller cache will be flushed to media, and 3. incoming FUA (Force Unit Access) host request will be ignored if a BBU is installed and enabled; Otherwise, will be honored. perform - provide the maximum performance and less data protection among the the controller settings. When user sets the storsave to perform mode, it means: 1. "Write Cache" will not be disabled when the unit becomes " DEGRADED ", 2. all data flushing from controller cache will be flushed to disk, and 3. incoming FUA (Force Unit Access) host request will be honored. When storsave is set to perform, a warning about the data loss in the event of power failure is giving to user to confirm the option. If user want to skip the confirmation, [quiet] option can be used to by pass the warning. balance - provide more data protection than perform mode but less data protection than protect mode. And provide better performance than protect mode but less performance than perform mode. When user sets the storsave to the balance mode, it means: 1. "Write Cache" will not be disabled when the unit becomes " DEGRADED ", 2. all data flushing from controller cache will be flushed to media if a BBU is installed and enabled; Otherwise, will be flushed to disk only, and 3. incoming FUA (Force Unit Access) host request will be ignored if a BBU is installed and enabled; Otherwise, will be honored. Example: localhost> /c0/u5 set storsave=protect
```

```
Setting Command Storsave Policy for unit /c0/u5 to [protect] ... Done. /cx/ux migrate type=RaidType [disk=p:-p] [group=3 | 4 | 5 | 6 | 7 | 8] [stripe=Stripe] [noscan] [nocache] [autoverify]
```

This command allows you to migrate an existing unit (aka source) to a unit with type=RaidType (aka destination) to achieve a goal of either capacity expansion. The destination is subject to similar rules and policies during its creation time. For example a valid number of disks and parameters must be specified. The destination unit must use all source disks and potentially augment in the disk=p:-p list for a larger capacity. Un-specified parameters are inherited from the source unit. 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 drives participating in RAID1 into multiple units of type single. The source unit name information will be duplicated to all destination units. But the source unit serial number will not be carried over to new unit. The new destination unit will have its own serial number.

This feature is only available with 9000 series of controllers.

Note: We do not recommend users to move a migrating unit to another controller before its migrate operation is completed. Any unclean shutdown on the migrating unit may lead the unit to a RECOVERY state in another controller which the migrate operation can not be continue until the unit is moved back to its original controller.

type=RaidType consists of the destination unit RAID type as in raid0, raid1, raid5, raid10, raid50, or single.

For example "type=raid5" indicates the destination unit is RAID-5 .

The following table illustrates valid migration paths:

Src/Dst	Raid0	Raid1	Raid5	Raid10	Raid50	Single	JBOD	Spare	-----+-----+-----+-----+-----+-----+-----+-----+-----+
Raid0	Y	N	Y	Y	Y	Y	N	N	N
Raid1	Y	N	Y	Y	Y	Y	Y	N	N

In the case of capacity expansion:

\$ tw_cli /c0/u0 migrate type=raid10 disk=3-4 stripe=16 Indicates that the destination unit has a RAID type of raid10 and has the disk 3, 4 in addition to all the disks in the existing unit u0.

The following is a mockup of how migrating units will be displayed...

3ware CLI> /c0 show Unit UnitType Status %Cmpl Stripe Size(GB) Cache AVerify IgnECC

u0 Migrator MIGRATING 67 - 149.05 ON OFF OFF u1 RAID-5 OK - 64k 298.22 ON OFF OFF 3ware CLI> /c0/u0 show Unit UnitType Status %Cmpl Port Stripe Size(GB) Blocks

u0 Migrator MIGRATING 67 - - - - su0 RAID-1 OK - - - 149.001 312477696 su0-0 DISK OK - p0 - 149.001 312477696 su0-1 DISK OK - p2 - 149.001 312477696

du0 RAID-10 OK - - 16K 298.002 624955392 du0-0 RAID-1 OK - - - - du0-0-0 DISK OK - p0 - 149.001 312477696 du0-0-1 DISK OK - p2 - 149.001 312477696 du0-1 RAID-1 OK - - - - du0-1-0 DISK OK - p3 - 149.001 312477696 du0-1-1 DISK OK - p4 - 149.001 312477696 The above report indicates that /c0/u0 is a migrating unit with 67% completion. The report also indicate that Source Unit su0 is of type RAID-1 and Destination Unit du0 is of type RAID-10 .

Port Object Messages

Port Object Messages are commands (a.k.a. methods/messages) that are sent to an instance of a disk which attaches to a port such as /c0/p0. /cx/px show

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

Example:

```
localhost> /c0/p5 show Port Status Unit Size Blocks Serial ----- p5 OK u5 149.05 GB 312581808 WD-
WMACK1406498 The above report indicate that port 5 of controller 0 is attached to one Western Digital disk with status OK participating in
unit 5. /cx/px show Attribute Attribute ... This command shows the current setting of the given attribute(s) on the specified port or disk. One
or many attributes can be requested. Invalid attribute will terminate the loop. Possible attributes are: capacity, firmware, identify(9550SX
and 9590SE only), lspeed(9550SX and 9590SE only), model, ncq(9550SX and 9590SE only), serial, smart, and status. /cx/px show status
This command presents the status of the specified port. Example: localhost> /c0/p5 show status /c0/p5 Status = OK /cx/px show model
```

This command presents the model of the specified port.

Example:

```
localhost> /c0/p5 show model /c0/p5 Model = WDC WD1600BB-00DAA0 /cx/px show serial This command presents the serial number of the
specified port. Example: localhost> /c0/p5 show serial /c0/p5 Serial = WD-WMACK1406498 /cx/px show firmware
```

This command presents the firmware version of the specified port.

Example:

```
localhost> /c0/p5 show firmware /c0/p5 Firmware Version = 65.13G65 /cx/px show identify (9550SX and 9590SE only) This command
presents the LED status of the port for 9550SX and 9590SE only. Example: localhost> /c0/p5 show identify /c0/p5 Identify Status = on /cx/px
show ncq (9550SX and 9590SE only)
```

This command presents the NCQ (Native Command Queueing) information of the port.

Example:

```
localhost> /c0/p5 show ncq /c0/p5 NCQ Supported = No /c0/p5 NCQ Enabled = No /cx/px show lspeed (9550SX and 9590SE only) This
command presents the SATA link speed supported by the disk behind the port and the actual speed is set to. Example: localhost> /c0/p5
show lspeed
```

```
/c0/p5 SATA Link Speed Supported = 3.0 Gb/s /c0/p5 SATA Link Speed = 3.0 Gb/s /cx/px show capacity
```

This command presents the capacity, both in human readable (such as GB) and block count of the specified port. Note that at this writing, human readable format is computed based on division by 1000 (not 1024) as is popular with hard disk vendors.

Example:

```
localhost> /c0/p5 show capacity 149.05 GB (312581808) /cx/px show smart This command extracts SMART (Self Monitoring Analysis and
Reporting) data from the specified disk. Note that this data is actually extracted live from the disk; as such this command could be used to
get most recent data about presence or lack of a disk. Be aware that extracting SMART data will burden the I/O bandwidth. Example:
localhost> /c0/p5 show smart
```

10 00 01 0B 00 C8 C8 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 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 00 2C Note that at this writing, we are not decoding the SMART data. Also note that if the disk attached to the specified port is not present or there are cabling problems reaching the disk, CLI will return an error. This could be one way of detecting whether a disk is present or not. /cx/px show all

This command shows the current setting of all above attributes.

```
/cx/px export [noscan] [quiet]
```

This command allows you to export (or remove) a port /cx/px (or drive). Exporting a port will instruct the firmware to remove the specified port from its pool of managed ports, but retains the DCB (Disk Configuration Block) meta-data on the attached disk. You can import (or re-introduce) the port via rescan. noscan is used to not inform the OS of this change. Default is to inform the OS. The quiet option is for non-interactive mode.

Use caution when using this command. Drives, which are part of a redundant array, can be removed, but the array will be degraded. Non-redundant drives, which are part of a unit, can not be removed. /cx/px set identify=on | off (9550SX and 9590SE only)

This command sets the LED status of the port. If the identify is set to on, the firmware activates the setting of the corresponding LED of the port. If the setting of the configuration for LED is blinking for identify=on, this will blink the LED of the port.

Note: Enclosure services hardware is also required.

Example:

```
localhost> /c0/p5 set identify=on Setting Port Identify on /c0/p5 to [on] ... Done. BBU Object Messages BBU (Battery Backup Unit) Object Messages are commands (a.k.a. methods/messages) that are sent to an instance of a BBU such as /c0/bbu. This object is only available on 9000S controllers where the BBU is actually installed. /cx/bbu show This command presents a summary report on the specified BBU object. For example: localhost> /cx/bbu show
```

```
Name OnlineState BBUReady Status Volt Temp Hours LastCapTest
```

bbu On No Testing OK OK 72 01-Jul-2004 Indicates the date the battery capacity was last measured is 01-Jul-2004. The battery is estimated to last for 72 hours from the last tested date. The BBU unit is currently testing the battery. Both voltage and temperature are normal. And the BBU is not ready to backup the write cache on the controller due to the testing. /cx/bbu show Attribute Attribute ...

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

```
/cx/bbu show status
```

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

Testing

Battery test is currently in progress. It 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 applicable RAID units are also disabled. If the test is completed with no error and the BBU returns back to WeakBat or OK state, the write cache will be resumed. If a Fault, Failed or an Error occurs during the test, the write cache remains at the disabled state until the problem is fixed.

Charging

BBU is currently charging the battery. The charging is started automatically by the BBU whenever necessary. During the charging, the BBU is not capable of backup operation and the write cache is disabled. Once charging is completed and the BBU returns back to OK status, the write cache will be resumed. If a FAULT or an ERROR occurs during the test, the write cache remains at the disabled state until the problem is fixed.

Fault

A battery fault is detected. At this state, the BBU is not capable of backup operation and the write cache is disabled. We recommend you to replace the battery and/or the BBU board to fix the problem as soon as possible so that the write cache will be enabled again.

Error

Other BBU error is detected. At this state, the BBU is not capable of backup operation and the write cache is disabled. We recommend you to replace the battery and/or the BBU board to fix the problem as soon as possible so that the write cache will be enabled again.

Failed

The battery failed a test. At this state, the BBU is not capable of backup operation and the write cache is disabled. We recommend you to replace the battery and/or the BBU board to fix the problem as soon as possible so that the write cache will be enabled again.

WeakBat

BBU is functioning normally which means it is online and capable of backing up the write cache. But the battery is weak and should be replaced.

OK

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

–

Battery is not present or BBU unit is not installed.

`/cx/bbu show batinst`

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

`/cx/bbu show lasttest`

This command shows the date the battery capacity was last measured.

`/cx/bbu show volt`

This command shows the voltage status of the battery. The status can be OK , HIGH , LOW , TOO-HIGH , and TOO-LOW . The HIGH and LOW are in warning range. TOO-HIGH and TOO-LOW are out of the operating range and need to be concerned.

`/cx/bbu show temp`

This command shows the temperature status of the battery. The status can be OK , HIGH , LOW , TOO-HIGH , and TOO-LOW . The HIGH and LOW are in warning range. TOO-HIGH and TOO-LOW are out of the operating range and need to be concerned.

`/cx/bbu show cap`

This command shows the battery capacity in hours.

`/cx/bbu show serial`

This command shows the BBU serial number.

`/cx/bbu show fw`

This command shows the BBU board firmware version number.

`/cx/bbu show pcb`

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

`/cx/bbu show bootloader`

This command shows the BBU 's Boot Loader version.

`/cx/bbu show all`

This command shows the current settings of all above attributes on the BBU board.

For example:

```
localhost> /cx/bbu show all /cx/bbu Firmware Version = BBU: 1.04.00.007 /cx/bbu Serial Number = Engineering Sample. /cx/bbu Online State = On /cx/bbu BBU Ready = Yes /cx/bbu BBU Status = OK /cx/bbu Battery Voltage = OK /cx/bbu Battery Temperature = OK /cx/bbu Estimated Backup Capacity = 241 Hours /cx/bbu Last Capacity Test = 22-Jun-2004 /cx/bbu Battery Intallation Date = 20-Jun-2004 /cx/bbu Bootloader Version = BBU 0.02.00.002 /cx/bbu PCB Revision = 65 localhost> /cx/bbu test [quiet]
```

This command starts the battery capacity test. The test may take up to 24 hours to complete. During the test, the BBU is not capable of backup operation and the write cache is disabled. The performance of all the units under the controller may be impacted because their write IOs are not cached. Once the test is completed with no error and the BBU returns back to OK state, the write cache will be resumed. The quiet option is for non-interactive mode.

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

Note: The test cannot be terminated before it completes. `/cx/bbu enable`

This command enables the BBU detection on the controller. And the controller will utilize BBU functionality in the event of power failure if BBU is there and ready.

`/cx/bbu disable [quiet]`

This command disables the BBU detection on the controller. In this situation, the controller ignores the existence of the BBU when the BBU detection is disabled. In another words, when a BBU attaches to a controller and the BBU detection is disabled, the storage management software will show user that there is no BBU installed on this controller. This is because the BBU detection is disabled. The quiet option is for non-interactive mode.

Help Commands

This command set provides brief on-line help. At top level of the command set, it is considered to be in the Shell Object. The Shell object has focus, show, flush, rescan, and commit commands. In addition to the Shell Object, we also have other objects such as /cx, /cx/ux, /cx/px, and /cx/bbu. Using the help command on objects, the help command shows all possible sub-commands associate with the object.

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

```
localhost> help /cx /cx show /cx show attribute [attribute ...] where attributes are: achip | allunitstatus | autocarve(9000 series) | bios | driver | firmware autorebuild(9550SX and 9590SE only) | carvesize(9000 series) drivestatus | exportjbod | ctlbus(9550SX and 9590SE only) memory | model | monitor | numdrives | numports | numunits | unitstatus ondegrade(9000S only) | pcb | pchip | serial | spinup | stagger /cx show all where all means attributes and configurations. /cx show diag /cx show alarms [reverse] /cx show rebuild (9000 series) /cx show verify (9000 series) /cx show selftest (9000 series) /cx add type=<RaidType> disk=<p:-p..> [stripe=<Stripe>] [noscan] [nocache] [group=<3 | 4 | 5 | 6 | 7 | 8>] [autoverify] [ignoreECC] [name=string (9000S only)] [storsave=<protect | balance | perform> (9550SX and 9590SE only)] RaidType = { raid0, raid1, raid5, raid10, raid50, single, spare, JBOD(7000/8000 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 exportjbod=on | off (9000 series) /cx set ondegrade=cacheoff | follow (9000S 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] (9000 series) /cx set autorebuild=on | off (9550SX and 9590SE 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 flush /cx commit ( Windows only ) (previously known as shutdown) /cx start mediascan (7000/8000 only) /cx stop mediascan (7000/8000 only) /cx rescan [noscan] NOTE: Does not import non-JBOD on 7000/8000 models. localhost> However, user can also get help with "?" or "help" as the user progresses with the specific object or a command after an object.
```

For example: User progresses to /c0 show and needs help on what specific attribute syntax, user can use '?' to get help as following:

```
localhost> /c0 show ? /cx show /cx show attribute [attribute ...] where attributes are: achip | allunitstatus | autocarve(9000 series) | bios | driver | firmware autorebuild(9550SX and 9590SE only) | carvesize(9000 series) drivestatus | exportjbod | ctlbus(9550SX and 9590SE only) memory | model | monitor | numdrives | numports | numunits | unitstatus ondegrade(9000S only) | pcb | pchip | serial | spinup | stagger /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) localhost> help
```

This command provide a table of contents, providing an overall navigational help. Typical output looks like:

```
localhost> help Copyright© 2004, 2005 Applied Micro Circuits Corporation(AMCC). All rights reserved. AMCC/3ware CLI (version 2.00.00.048) Commands Description ----- info Displays information about controller(s), unit(s) and port(s). maint Performs maintenance operations on controller(s), unit(s) and ports. alarms Displays current AENs. set Displays or modifies controller and unit settings. sched Schedules background tasks on controller(s) (9000 controllers only). quit Exits the CLI. --- Primary Command Syntax --- 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). commit Commit dirty DCB to storage on controller(s). /cx Controller specific commands. /cx/ux Unit specific commands. /cx/px Port specific commands. /cx/bbu BBU specific commands. (9000 controllers only) Type help <command> to get more details about a particular command. For more detail information see tw_cli's documentation. localhost> 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 "Shell Object Messages" section for more on show.

help flush

This command provides specific flush related help, illustrating various ways to use the flush command. See the "Shell Object Messages" section for more.

help rescan

This command provides specific rescan related help, illustrating various ways to use the rescan command. See the "Shell Object Messages" section for more.

help commit

This command provides specific commit related help, illustrating various ways to use the commit command. See the "Shell Object Messages" section for more.

help focus

This command provides specific focus related help, illustrating various ways to use the focus command. See the "Shell Object Messages" section for more.

help /cx

This command provides specific controller /cx related help, illustrating various commands associated with the controller /cx. See the "Controller Object Messages" section for more.

help /cx/ux

This command provides specific unit /cx/ux related help, illustrating various commands to use on a unit /cx/ux. See the "Controller Object Messages" section for more.

help /cx/px

This command provides specific /cx/px related help, illustrating various ways to use the /cx/px command. See the "Port Object Messages" section for more.

help /cx/bbu

This command provides specific /cx/bbu related help, illustrating various ways to use the /cx/bbu command. See the "BBU Object Messages" section for more.

Legacy or Original Command Syntax (For a Limited Time) The functionality provided by tw_cli(8) in the legacy command syntax can be divided into the following categories:

Important Notice: The legacy command syntax will be discontinued in the future releases. Please change your existing CLI related scripts or automation programs with the primary command syntax as soon as possible. info

A collection of sub-commands under this category allows you to view attributes of objects such as controllers, logical units and drives. This is a read-only operation.

maint

Sub-commands under this category allow you to create and mutate objects and their attributes such as creating and deleting logical units, rebuilding, flushing, etc. These commands are read/write operations and should be used with care.

sched

This is a read/write operation. It allows you to set background tasks associated with a controller. Background tasks are executed concurrent to I/O and management operations.

alarms

Displays alarms, which have been generated by controllers. Alarms or AENs (Asynchronous Event Notification) have different levels of severity. These can be extracted and archived for overall trend analysis.

set

Displays or modifies controller settings such as enabling or disabling cache on units, turning auto-verify on or off.

help

Display help on various sub-commands.

Info Commands

Info commands are read-only operations showing various values of controllers, units and drives. info

Provides information on all detected controllers.

info cid

Provides an overall summary information on controller cid.

info cid driver

This command reports the device driver version associated with controller cid.

Example:

localhost> info c0 driver /c0 Driver Version = 1.02.00.036 info cid model This command reports the controller model of controller cid. Example: localhost> info c0 model /c0 Model = 7500-12 info cid firmware

This command reports the firmware version of controller cid.

Example:

localhost> info c0 firmware /c0 Firmware Version = FGXX 2.01.00.025 info cid bios This command reports the BIOS version of controller cid. Example: localhost> info c0 bios /c0 BIOS Version = BG9X 2.01.00.026 info cid monitor

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

Example:

localhost> info c0 monitor /c0 Monitor Version = BLDR 1.00.00.008 info cid serial This command reports the serial number of the specified controller cid. Example: localhost> info c0 serial /c0 Serial Number = F12705A3240009 info cid pcb

This command reports the PCB (Printed Circuit Board) revision of the specified controller cid.

Example:

localhost> info c0 pcb /c0 PCB Version = Rev3 info cid pchip This command reports the PCHIP (PCI Interface Chip) version of the specified controller cid. Example: localhost> info c0 pchip /c0 PCHIP Version = 1.30-33 info cid achip

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

Example:

localhost> info c0 achip /c0 ACHIP Version = 3.20 info cid numports This command reports the port capacity (number of physical ports) of the specified controller cid. Example: localhost> info c0 numports /c0 Number of Ports = 12 info cid numunits

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

Example:

localhost> info c0 numunits /c0 Number of Units = 1 info cid numdrives This command reports the number of drives currently managed by the specified controller cid. This report does not include (logically) removed/exported drives. Also note that physically removed disk(s) will not be detected unless I/O is performed against the disk. See info cid pid smart for a workaround. Example: localhost> info c0 numdrives /c0 Number of Drives = 5 info cid unitstatus

This command presents a list of units, their types, capacity and status currently managed by the specified controller cid.

Example:

localhost> info c0 unitstatus Unit UnitType Status %Cmpl Stripe Size(GB) Cache AVerify IgnECC ----- u0 RAID-5 OK - 64K 223.485 OFF OFF ON u1 JBOD OK - - 76.3352 OFF OFF - u2 RAID-0 OK - 64K 148.99 ON ON - info cid allunitstatus This command presents a count of total and Not Optimal units managed by the specified controller cid. See "Info Commands" for more on Not Optimal definition. Example: localhost> info c0 allunitstatus /c0 Total Optimal Units = 2 /c0 Not Optimal Units = 0 info cid drivestatus

This command presents a list of drives, port assignment, vendor signature, size, status, and unit membership/affiliation.

Example:

localhost> info c0 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 - - - - info cid uid This command presents detailed information on the specified unit uid. 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. One application of this 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. Example: localhost> info c0 u0

Unit UnitType Status %Cmpl Port Stripe Size(GB) Blocks

u0 RAID-50 OK - - 64K 596.05 1249921024 u0-0 RAID-5 OK - - 64K - - u0-0-0 DISK OK - p0 - 149.10 312481280 u0-0-1 DISK OK - p2 - 149.10 312481280 u0-0-2 DISK OK - p3 - 149.10 312481280 u0-1 RAID-5 OK - - 64K - - u0-1-0 DISK OK - p4 - 149.10 312481280 u0-1-1 DISK OK - p5 - 149.10 312481280 u0-1-2 DISK OK - p6 - 149.10 312481280 info cid uid status

This command presents the status of the specified unit.

Example:

localhost> info c0 u0 status /c0/u5 status = OK info cid uid rebuildstatus This command presents the rebuildstatus (if any) of the specified unit. Example: localhost> info c0 u5 rebuildstatus /c0/u5 is not rebuilding, its current state is OK info cid uid verifystatus

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

Example:

```
localhost> info c0 u5 verifystatus /c0/u5 is not verifying, its current state is OK info cid uid initializestatus This command presents the initializestatus (if any) of the specified unit. Example: localhost> info c0 u5 initializestatus /c0/u5 is not initializing, its current state is OK info cid pid
```

This command presents various information on the specified disk attached to port `pid`. Typical information looks like:

Example:

```
localhost> info c0 p5 Port Status Unit Size Blocks Serial ----- p5 OK u5 149.05 GB 312581808 WD-WMACK1406498 The above report indicate that port 5 of controller 0 is attached to one Western Digital disk with status OK participating in unit 5. info cid pid status This command presents the status of the specified port. Example: localhost> info c0 p5 status /c0/p5 Status = OK info cid pid model
```

This command presents the model of the specified port.

Example:

```
localhost> info c0 p5 model /c0/p5 Model = WDC WD1600BB-00DAA0 info cid pid serial This command presents the serial number of the specified port. Example: localhost> info c0 p5 serial /c0/p5 Serial = WD-WMACK1406498 info cid pid firmware
```

This command presents the firmware version of the specified port.

Example:

```
localhost> info c0 p5 firmware /c0/p5 Firmware Version = 65.13G65 info cid pid capacity This command presents the capacity, both in human readable (such as GB ) and block count of the specified port. Note that at this writing, human readable format is computed based on division by 1000 (not 1024) as is popular with hard disk vendors. Example: localhost> info c0 p5 capacity /c0/p5 Capacity = 149.05 GB (312581808) info cid pid smart
```

This command extracts SMART (Self Monitoring Analysis and Reporting) data from the specified disk. Note that this data is actually extracted live from the disk, as such this command could be used to get most recent data about presence or lack of a disk. Be aware that extracting SMART data will burden the I/O bandwidth.

Example:

```
localhost> info c0 p5 smart 10 00 01 0B 00 C8 C8 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 00 00 00 2C Note that at this writing, we are not decoding the SMART data. Also note that if the disk attached to the specified port is not present or there are cabling problems reaching the disk, CLI will return an error. This could be one way of detecting whether a disk is present or not. info cid 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 this output to a file, where it can be communicated to tech support or further studied with Linux utilities like od(1). Example: $ tw_cli info c0 diag > diag.txt info cid exportjbod (9000 series) This command presents the current JBOD Export Policy; "on", "off" or "Not Supported". Example: localhost> info c0 exportjbod /c0 JBOD Export Policy = Not Supported.
```

```
localhost> info c1 exportjbod /c1 JBOD Export Policy = on Maint Commands Sub-commands under this category allow you to create and mutate objects and their attributes such as creating and deleting logical units, rebuilding, flushing, etc. These commands are read/write operations and should be used with care. maint rescan [cid ...] [noscan] This command instructs the controller to rescan all ports and reconstitute all units. The controller will update its list of ports (attached disks), and visits 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. If no controller is specified, all controllers will be rescanned. One or many controllers can be specified. noscan is used to not inform the OS of the unit discovery. Default is to inform the OS . Example: localhost> maint rescan Rescanning controller /c0 for units and drives ...Done. Found following unit(s): [none]. Found following drive(s): [none]. Rescanning controller /c1 for units and drives ...Done. Found following unit(s): [/c1/u3]. Found following drive(s): [/c1/p7, /c1/p8]. Note: Does not import non-JBOD on 7000/8000 models
```

```
maint remove cid uid [noscan]
```

This command allows you to remove (or export) a unit. Exporting a unit will instruct the firmware to remove the specified unit from its pool of managed units, but retains the DCB (Disk Configuration Block) meta-data. As such the unit can later be imported back. `noscan` is used to not inform the OS of this change. Default is to inform the OS . Use caution when using this command. Units that are currently in use or mounted cannot be removed.

```
maint remove cid pid [noscan]
```

This command allows you to remove (or export) a port (or drive). Exporting a port will instruct the firmware to remove the specified port from its pool of managed ports, but retains the DCB (Disk Configuration Block) meta-data on the attached disk. You can import (or re-introduce) the port via `rescan`. `noscan` is used to not inform the OS of this change. Default is to inform the OS . Use caution when using this command. Drives, which are part of a redundant array, can be removed, but the array will be degraded. Non-redundant drives, which are part of a unit, can not be removed.

maint deleteunit cid uid [noscan]

This command allows you to delete a unit. Deleting a unit not only remove the specified unit from the controller’s list of managed units, but also destroys the DCB (Disk Configuration Block) meta–data. Ports (or disks) associated with this unit will now be part of the free pool of managed disks. This is a destructive command and should be used with care. noscan is used to not inform the OS of this change. Default is to inform the OS . Use caution when using this command. Units that are currently in use or mounted cannot be removed.

maint createunit cid RaidType PidList [Stripe] [noscan] [DskGrp] [nocache] [autoverify] [ignoreECC]

This command allows you to create a unit on the specified controller cid, of type RaidType, optional stripe size of Stripe, using one or many disks specified by PidList. By default the host operating system will be informed of the new block device and write cache is enabled. In case of RAID–50 , you can also specify the layout of the unit by specifying the number of disks per diskgroup with DskGrp option.

Since this command is by far the richest command, it deserves more details.

cid is the controller name as in c0, c1, etc.

RaidType consists of letter “r” followed by RAID or logical unit type as in raid0, raid1, raid5, raid10, raid50, single, spare, and jbod. For example “rraid50”. The following table illustrates supported types and controller models.

Model | R0 | R1 | R5 | R10 | JBOD | Spare | R50 | Single | ---+---+---+---+---+---+---+---+---+---+ 7K/8K | Y | Y | Y | Y | Y | Y | Y | N | N | ---+---+---+---+---+---+---+---+---+---+ 9K | Y | Y | Y | Y | Y | Y | Y | Y | PidList consists of letter “p” followed by 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 “:” or “-” as port index separators. A dash indicates a range and can be mixed with “:”. For example p0:1:2–5:9:12 indicates port 0, 1, 2 thru 5 (inclusive), 9 and 12.

Stripe consists of letter “k” followed by 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 (kilo bytes).

Model | R0 | R1 | R5 | R10 | JBOD | Spare | R50 | Single | ---+---+---+---+---+---+---+---+---+---+ 7K/8K | 64 | N/A | 64 | 64 | N/A | N/A | N/S | N/S |

	128			128					
	256			256					
	512			512					
	1024			1024					

---+---+---+---+---+---+---+---+---+---+ 9K | 16 | N/A | 16 | 16 | N/A | N/A | 16 | N/A |

	64		64	64			64		
	256		256	256			256		

---+---+---+---+---+---+---+---+---+---+ DskGrp consists of letter “g” followed by the number of disks per group for a RAID–50 type. Recall that a RAID–50 is a multi-tier array. At the most bottom layer, N number of disks per group are used to form the RAID–5 layer. These RAID–5 arrays are then integrated into a RAID–0 . This attribute allows you to specify the number of disks in the RAID–5 level. Valid values are 3, 4, 5 and 6.

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

The default gDskGrp varies based on number of disks. For 6 & 9 disks, the default is 3. For 8 disks, the default is 4. For 10 disks, the default is 5 and for 12 disks, the default is 4. Since 12 disks could be grouped into 3, 4, or 6 RAID–5 arrays. A grouping of 4 was set by default as it provides best net capacity and performance.

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

nocache attribute instructs CLI disable the write cache on the newly create unit. Enabling write cache increases performance at the cost of high–availability. A BBU (battery backup unit) or UPS is recommended to prevent data lost if the cache is enabled in the event of a power failure.

autoverify attribute enables the autoverify attribute on the unit that is to be created. For more details on this feature, refer to “Set Commands” section of this document. This feature is not supported on model 7000/8000. On model 9000, JBOD autoverify attribute is not persistent (does not survive reboots).

ignoreECC attribute enables the ignoreECC, also known as the overwriteECC attribute on the unit that is to be created. For more details on this feature, refer to “Set Commands” section of this document. The following table illustrates the supported RAID unit types. This table only applies to setting this feature at unit creation time. Generally ignoreECC applies to redundant units.

Model | R0 | R1 | R5 | R10 | JBOD | Spare | R50 | Single | ---+---+---+---+---+---+---+---+---+---+ 7K/8K | N | N | N | N | N |

N | N | N | ---+---+---+---+---+---+---+---+---+ 9K | N | Y | Y | Y | N | N | Y | N | maint rebuild cid uid pid [ignoreECC]

This command allows you to rebuild a DEGRADED unit by using the specified port. Rebuild only applies to redundant arrays such as RAID-1 , RAID-5 , RAID-10 and RAID-50 . During rebuild, bad sectors on the source disk will cause the rebuild to fail. You can allow for the operation to continue via the ignoreECC attribute. Rebuild process is a background task and will change the state of a unit to REBUILDING . Various info commands also show a percent completion as rebuilding progresses.

Note that the port (disk) to be used to rebuild a unit, must be a SPARE or unconfigured disk.

maint rebuild cid uid pause (7/8000 controllers only)

This command allows you to pause the rebuild operation on the specified unit. This feature is intended for model 7000 and 8000 only. Model 9000 has an on-board scheduler where rebuild operations can be scheduled to take place at specified start and stop times. Rebuild pause and resume function is provided to enable 7000/8000 users to achieve similar functionality with use of OS provided schedulers such as cron(8) or, at(1) in linux or user supplied programs.

See also "Sched Commands".

maint rebuild cid uid resume (7/8000 controllers only)

This command allows you to resume the rebuild operation on the specified unit. See "rebuild cid uid pause" for more.

maint flush cid [uid ...]

This command allows you to flush the write cache on the specified unit or all units associated with controller cid. Note that this command does not apply to spare unit types.

maint verify cid uid [stop]

This command starts or stops a background verification process on the specified unit uid. The following shows the supported matrix 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 that controller model.

Model | R-0 | R-1 | R-5 | R-10 | R-50 | Single | JBOD | Spare | ---+---+---+---+---+---+---+---+---+ 7K/8K | No | Yes | Yes | Yes | N/A | N/A | No | No | ---+---+---+---+---+---+---+---+---+ 9K | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Note that if subsequent to this command, one enables the background verify task to follow the scheduled slots, then this on-demand task will be paused until the scheduled time.

maint mediascan cid start | stop (7/8000 controllers only)

This command applies to 7000/8000 controllers. It provides media scrubbing for validating disk functionality. This includes bad block detection and remapping, etc. The command starts or stops a media scan operation on the specified controller cid.

maint commit cid

This command instructs the controller to commit its dirty DCBs to persistent storage (i.e. disks). While controller is processing I/O requests against underlying disks, an in-transaction bit is set. If a failure (such as power failure) is experienced, subsequent read from the disks, will inform the controller that an un-clean shutdown took place. This command allows the end user to complete all pending I/Os on disk(s) and clear the in-transaction bit.

Typical application of this feature is when an application is using a given unit in raw mode (such as databases) and user would like to shutdown the host (Including UPS post failure automations). This command can then expedite the process by instructing the controller to finish pending requests, clear DCB 's in-transaction flag as we are going down.

This command only applies to Windows operating system.

Sched Commands

Model 9000 supports background tasks. Background tasks include "rebuild", "verify", or "self test" activities. For each activity, up to 7 tasks can be registered, known as slots 1 through 7. Each activity can be managed by this command including adding, removing, enabling and disabling a task. Background tasks have a slot id, category (rebuild, verify, selftest), start-day-time and duration attributes.

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 window, if enabled.

verify activity attempts to verify all units based on their unit type. Verifying RAID-1 involves checking that both drives contain the exact data. On RAID-5 , the parity information is used to verify data integrity. RAID-10 and 50 are composite types and follow their respective array types. On 9000 series, non-redundant units such as RAID-0 , JBOD , single, and spare, are also verified (by reading and reporting unreadable sectors).

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

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

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

This command displays the current rebuild background tasks as illustrated below.

```
$ tw_cli sched rebuild c1
```

```
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 - - - - 4 - - - - 5 - - - - 6 Mon 1:00am 4 hr(s) disabled 7 Sun 12:00am 1 hr(s) disabled Status "disabled" indicates that the controller will not use the tabled schedules.
```

```
sched rebuild cid add day hour duration
```

This command adds a new background rebuild task to be executed on day (range 0 .. 6, where 0 is Sunday), at hour (range 0 .. 23), for a duration of duration (range 1 .. 24) hours. This command will fail if no (empty) slot is available.

For example:

```
$ tw_cli sched rebuild c1 add d0 h16 t3 Will add a rebuild background task to be executed on Sundays at 4:00 PM for a duration of 3 hours.
```

```
sched rebuild cid remove slot_id
```

This command will remove (or unregister) the rebuild background task in slot slot_id.

For example:

```
$ tw_cli sched rebuild c1 remove 2 Will remove rebuild background task in slot 2.
```

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

```
sched rebuild cid enable
```

This command will enable ALL rebuild background tasks on controller cid.

```
sched rebuild cid disable
```

This command will disable ALL rebuild background tasks on controller cid.

```
sched verify cid
```

This command displays the current verify background task as illustrated below.

```
$ tw_cli sched verify c1
```

```
Verify Schedule for Controller /c1
```

```
Slot Day Hour Duration Status
```

```
1 Mon 2:00am 4 hr(s) disabled 2 - - - - 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 Status indicates that the controller will not use the tabled schedules.
```

```
sched verify cid add day hour duration
```

This command adds a new background verify task to be executed on day (range 0 .. 6, where 0 is Sunday), at hour (range 0 .. 23), for a duration of duration (range 1 .. 24) hours. This command will fail if no (empty) slot is available.

For example:

```
$ tw_cli sched verify c1 add d0 h16 t3 Will add a verify background task to be executed on Sundays at 4:00 PM for a duration of 3 hours.
```

```
sched verify cid remove slot_id
```

This command will remove (or unregister) the verify background task in slot `slot_id`.

For example:

```
$ tw_cli sched verify c1 remove 3
```

Will remove rebuild background task in slot 3.

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

```
sched verify cid enable
```

This command will enable ALL verify background tasks on controller `cid`. When enabled, only tabled scheduled tasks will be followed (or used). Any previous on-demand background tasks will be ignored.

```
sched verify cid disable
```

This command will disable ALL verify background tasks on controller `cid`.

```
sched selftest cid
```

This command displays the current selftest background task as illustrated below.

```
$ tw_cli sched selftest c1
```

```
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 sched selftest cid add day hour
```

This command adds a new background selftest task to be executed on day (range 0 .. 6, where 0 is Sunday), at hour (range 0 .. 23). Notice that selftest runs to completion and as such no duration is provided. This command will fail if no (empty) slot is available.

For example:

```
$ tw_cli sched selftest c1 add d0 h16
```

Will add a selftest background task to be executed on Sundays at 4:00 PM .

```
sched selftest cid remove slot_id
```

This command will remove (or unregister) the selftest background task in slot `slot_id`.

For example:

```
$ tw_cli sched selftest c1 remove 3
```

Will remove rebuild background task in slot 3.

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

```
sched selftest cid enable [selftest_task_id]
```

This command will enable a particular `selftest_task` (UDMA or SMART). While `s0` is interpreted as UDMA , `s1` is interpreted as SMART . If no `selftest_task_id` is specified, both tasks will be enabled.

For example:

```
$ tw_cli sched selftest c1 enable s0
```

Will enable UDMA selftest on controller `c0`.

```
sched selftest cid disable [selftest_task_id]
```

This command will disable a particular `selftest_task` (UDMA or SMART). While `s0` is interpreted as UDMA , `s1` is interpreted as SMART . If no `selftest_task_id` is specified, both tasks will be disabled.

For example:

```
$ tw_cli sched selftest c1 disable s1
```

Will disable SMART selftest on controller `c0`.

Alarms Commands

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

Alarms generated on 7000/8000 models do not have dates, as such you'll see a '-' (read not-applicable) in "Date" column. Also on

7000/8000 models, the alarm message, contain the severity as well, hence the "Severity" column is showing a '-' as well. alarms [cid]

This command displays all available alarms on a given controller. Invoked without any controller id, will display alarms associated with all detected controllers.

Typical output looks like:

```
localhost> alarms Ctl Date Severity Message ----- c0 - - ERROR: Unit degraded: Unit #0 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 Set Commands These commands allow you to view and/or set certain controller and unit specific parameters as
described below. set rebuild cid 1..5 This command allows you to set priority of rebuild vs I/O operations. Setting this value to 1 implies that
rebuilds should consume more resources (cpu time, I/O bandwidth) to complete its task. Conversely setting this value to 5 implies that I/O
has higher priority than rebuild. This command applies to 7000, 8000, and 9000 models. For 7/8000 series, the rebuild rate also applies to
verify and mediascan tasks. set verify cid 1..5 This command allows you to set priority of verification vs I/O operations. Setting this value to
1 implies fastest verify, and 5 implies fastest I/O. Note that this feature only applies to 9000 models. set cache cid uid on|off This command
allows you to turn the write cache on a specified unit, on or off. This feature is supported on both 7000/8000 and 9000 models. The following
is the RAID Type vs Model support matrix. Model | R-0 | R-1 | R-5 | R-10 | R-50 | Single | JBOD | Spare |
---+---+---+---+---+---+---+---+---+---+---+---+ 7K/8K | Yes | Yes | Yes | Yes | N/A | N/A | Yes | No |
---+---+---+---+---+---+---+---+---+---+---+---+ 9K | Yes | Yes | Yes | Yes | Yes | Yes | Yes | No |
---+---+---+---+---+---+---+---+---+---+---+---+ set autoverify cid uid on|off This command allows you to turn on/off the autoverify
operation on a specified unit during allocated windows specified by sched verify commands. The autoverify operation is a continuous verify
operation, which takes place within the window specified with "sched verify". While the "sched verify" command allows you to specify the
time for the verify operation, this command allows you to enable or disable the autoverify operation on the specified unit. This feature only
applies to 9000 models. set overwriteECC cid uid on|off This command allows you to set the ignoreECC policy for a given unit such that
during rebuild of the specified unit, which could begin automatically (if the unit is degraded and spare has been defined) or manually, to be
applied to the rebuild operation. Setting overwriteECC to on means ignoreECC. This feature only applies to 9000 models. Help Commands
This command set provides brief on-line help. help This command provide a table of contents, providing an overall navigational help. Typical
output looks like: localhost> help Copyright© 2004, 2005 Applied Micro Circuits Corporation(AMCC). All rights reserved. AMCC/3ware CLI
(version 2.00.00.048) Commands Description
```

info Displays information about controller(s), unit(s) and port(s). maint Performs maintenance operations on controller(s), unit(s) and ports. alarms Displays current AENs. set Displays or modifies controller and unit settings. sched Schedules background tasks on controller(s) (9000 controllers only). quit Exits the CLI.

1. — Primary Command Syntax —

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). commit Commit dirty DCB to storage on controller(s). /cx Controller specific commands. /cx/ux Unit specific commands. /cx/px Port specific commands. /cx/bbu BBU specific commands. (9000 controllers only)

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

This command provides specific info related help, illustrating various ways to use the info command. Info provides reports on Controllers, Units and Drives. See the "Info Commands" section for more on info.

help alarms

This command provides specific alarms related help, illustrating various ways to use the alarms command. See the "Alarms Commands" section for more.

help set

This command provides specific set related help, illustrating various ways to use the set command. See the "Set Commands" section for more.

help maint

This command provides specific maintenance related help, illustrating various ways to use the maint command. See the "Maint Commands" section for more.

help sched

This command provides specific scheduling related help, illustrating various ways to use the sched command. See the "Sched Commands" section for more.

help quit

This command provides specific quit related help, illustrating various ways to use the quit command. See the "Quit Commands" section for more.

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.

ERRATA Meta-Character Warning:

If you wish to use CLI in single command mode (not interactive), make sure to avoid collision with your command interpreter (OS shell) by escaping the meta-characters (such as ?, <, >, @, &, *, etc) appropriately with single quote around them.

For example, given the

```
$ tw_cli /c0 ?
```

This is a case of single command usage where the user intends to get help on Controller related commands. While this is a valid CLI command, but since the arguments to CLI are first processed by the shell, then some shells like csh(1) will interpret the '?' as a meta-character to be used toward file completion and if no file is found with a single character, then shell will complain before the arguments are even passed down to CLI .

One solutions of this problem can be :

```
$ tw_cli help /cx
```

or

```
$ tw_cli '/c0 ?'
```

Note: Some of the OS shell does not have this problem such as bash.

Reporting Style

tw_cli(8) reporting has changed (hopefully for better). The intent has been to provide a consistent tabular reporting so that relevant and important information (such as info) are made available as fast as possible. For example, firmware, PCB , PCHIP and similar information have been removed from the info summary report, as this type of information is not frequently needed.

The new style also accommodates automation much better by providing consistent columns with or without values so that it could be easily parsed. The intent is to make CLI yet another API (to approach it).

However to accommodate current automations around tw_cli and to ease the migration, the old behavior can still be requested by setting TW_CLI_STYLE environment variable to OLD as follows:

If Bash, then "export TW_CLI_STYLE=OLD" If csh, then "setenv TW_CLI_STYLE OLD" if Windows, then "set TW_CLI_STYLE=OLD" This backward compatibility window, will be communicated by official AMCC/3ware representatives.

Initialization Process Control

On the 9K series of controllers, the rebuild scheduling controls both rebuild and initialize processes if it is enabled. Currently, tw_cli(8) does not have any direct command to pause or resume an initialization process. If such action is needed, use the rebuild scheduling to handle it.

ENVIRONMENT VARIABLES TW_CLI_STYLE setting this variable to OLD , will provide the old reporting style. TW_CLI_INPUT_STYLE setting this variable to OLD , will disable focus feature in the interactive mode.

AUTHOR Medi Montaseri

SEE ALSO AMCC/3ware CLI User Guide AMCC/3ware User Guide AMCC/3ware Installation Guide <http://www.amcc.com>

From:
<https://wiki.nanoscopic.de/> - **nanoscopic wiki**

Permanent link:
https://wiki.nanoscopic.de/doku.php/pages/man/tw_cli?rev=1612004401

Last update: **2021/01/30 11:00**

