

# **QShell Command Reference Manual**

---

---

**QSR Version – 3.1**

**Document Version 1.0.2**

**August 2005**

# TABLE OF CONTENTS

Finding Information.....	4
About this manual .....	4
Supported Products .....	4
How this manual is organized .....	4
QShell overview .....	5
Security Levels.....	5
Command Prompt.....	5
Command History .....	6
Command Types.....	6
SCSI device ID table.....	7
QShell system commands.....	8
ACCOUNT .....	9
ASSIGN.....	11
BLDCLIP .....	12
CAPTURE .....	13
CD .....	13
CLIPTC.....	14
CLOCK.....	15
COMMENT.....	16
CONFIG .....	16
CONNCTRL .....	17
COPY .....	18
DEL .....	21
DELTREE.....	22
DIR .....	22
DRVMAP .....	23
DUMP .....	24
EXIT .....	25
FS ARRAY .....	26
FS FORMAT .....	27
FS MIRRORSETUP.....	30
GETTC .....	31
HDDIAG .....	32
HELP .....	34
INFO.....	34
JOG.....	35
LICENSE.....	36
LIST.....	37
LOG.....	38
LOGIN .....	40
LOGOUT .....	40
MIRROR.....	41
MkDIR .....	42
MODCLIP.....	42
MOUNT .....	45
NET .....	46
PATH.....	47
PLAY .....	48
PLAYSCRIPT .....	50
POSTDUB.....	51

PRINT.....	54
PULLAUD.....	54
PWD.....	56
QUEUE.....	56
QQDOWNLOAD .....	56
F2Q - Receiving a file from a computer workstation.....	57
QQUPLOAD.....	58
Q2F – Sending a file to a computer workstation.....	59
RECORD.....	60
RECOVER .....	64
REINDEX .....	64
REN.....	65
RENVOL.....	65
REPAIR.....	66
RESET .....	67
SCSILIST .....	67
SERCTRL .....	68
SET .....	71
SHELL.....	72
SHOW .....	72
SHUTDOWN .....	73
SHUTTLE.....	74
STAT .....	74
STOP.....	74
TAPE.....	75
TEMP .....	80
TIMELINE.....	81
TYPE.....	83
UNMOUNT.....	83
VARSPPEED .....	84
VER.....	84
VMODE .....	85
VOLUTIL .....	86
VTPMERGE .....	88
WAIT .....	90
WHATIS .....	90
WHICH .....	91

# Finding Information

## About this manual

This reference manual contains supplementary information for operating QuVIS video server products from the command line interface, QShell.

## Supported Products

The QShell command line interface is a feature found on all QuVIS video servers. This document covers the available QSR 3.1 QShell commands for the most recent QuVIS products including:

- QuVIS Acuity
- QuVIS Ovation
- QuVIS Encore
- QuVIS Cinema Player

This document details all of the available QShell commands that may be run/executed on the above listed products. While most of the QShell commands are available to all products there is one notable exception.

- 1.** The QuVIS Ovation and QuVIS Cinema Player products are player-only products. These units do not have the capability to record content. For this reason the following commands are not supported by these products:
  - a.** Record
  - b.** Vmode R

## How this manual is organized

This manual is divided into the following two sections:

[QShell overview](#) – This section provides an introduction to the QShell environment.

[QShell system commands](#) – This section provides a description of each QShell command, outlines its command structure and provides practical examples on how to use each command.

## QShell overview

QShell is the command line interface or shell that may be used to communicate with and control the system using a PC, edit or show controller. This command interface is a powerful tool used for diagnostics and advanced functions that are not available from the front panel menu system.

The QShell interface is always available on RS232 port A (9-pin connector) located on the front of the unit to the right of the power button. For RS232 devices, you need an “RS-232 null modem” serial cable.

Alternately, a QShell session may also be remotely loaded or accessed using a Telnet terminal program on a PC. In this case, only network connectivity between the PC and the QuVIS video server is required.

**NOTE:** RS-232 and RS-422 serial connections are not used to transfer data between the unit and your computer or network; they are used for remote control of the system only. (An Ethernet connection is used for data transfers, along with an FTP program or QuRemote).

## Security Levels

If user security is enabled, the availability of some QShell commands is dictated by user permissions. The minimum-security group required for each command is listed in the far right column of each command title. If user security is not enabled, then user permissions default to supervisor level.

## Command Prompt

The QShell prompt is always the current directory followed by the prompt symbol: >. All commands in this shell are specific to QuVIS Soft Release (QSR) 3.x. They resemble “typical” DOS and UNIX shell commands, and so should be familiar if you have experienced these systems before.

The QShell command prompt displays the volume and the current working directory followed by a greater-than (>) sign. The current path is used by most QShell commands as the default source for their operation. For example, if you issue a *PLAY* command and specify a clip name without specifying the full path, the command will expect the specified clip to exist in the current directory. The source or destination file for a command does not need to be located in the current directory. To reference a source file or destination location, you must use a valid path when executing the command.

Example 1 – QShell command prompt (root)

```
Volume:>
```

Example 2 – QShell command prompt (path)

```
Volume:clips>
```

To specify a different path while executing a QShell command there are a couple of methods that can be used.

1. Use a preceding slash “/” to indicate that the following path begins at the root of the current volume.

Example:        `Volume:project>play /clips/sampleclip`

2. To specify a directory path that is subdirectory of the current path, specify the subdirectory without a preceding slash.

Example:        `Volume:>play clips/sampleclip`

3. To specify a path that begins back one directory or in the parent directory, use the “..” method depicted below. In the below example, the file to be played (sampleclip) is located in the “clips” directory on the current volume.

Example:           Volume:clips/feature>play ../sampleclip

4. If the source or destination path is located on a separate volume, the volume is indicated by using the colon “:” volume designator after the volume name.

Example:           Volume:>play Transport:clips/sampleclip

## Command History

The QShell maintains a 20-line history buffer. By using the up/down arrow keys, you can browse through the past 20 commands that have been entered on the command line.

For this feature to work, your terminal must send ANSI sequences in response to up/down arrow keys. Typically this requires VT-100 terminal emulation.

**TIP:** Use the **TAB** key to complete paths and file names. While typing in a file name, type in the first couple of characters and then press the TAB key. If there are multiple files with the same prefix, keep pressing the TAB key until the desired file is auto-completed.

## Command Types

There are three general types of commands that may be executed from the command line interface: internal commands, software modules and batch files.

### System commands

Allow you to perform built-in functions like copying a file or listing the contents of a directory. Available internal commands are listed when you use the *help* command. The *help* command does not list executable software modules or batch files.

### Software modules

Software modules are actually small utility programs that can be run from the command line. System software modules are located in the *sys/mod* directory. You do not need to include the .mod file extension when invoking one of these commands.

### Batch Files

Batch files are text files with a “.bat” file extension. Generally these files are placed in the *user/bat* directory. Batch file may contain any command that you would use at the QShell, including internal commands, software modules and other batch files. Additionally, batch files may be executed from the front panel menu system for custom applications.

Arguments may be passed batch files. Arguments work just like standard DOS batch file arguments: in the batch file, use %1 to refer to the first argument, %2 for the second, etc. The argument %0 can be used for the name of the batch file that was invoked. Some improvements over DOS: if an argument is referenced in the batch file text that was not provided at the batch file’s invocation, an error is reported, and a NULL string is used. For even better checking, place a line such as .3 at the very top of the batch file. This specifies the required number of arguments that must be passed to the file, otherwise an error is reported and none of the file is executed.

Comments may be placed in batch files by using a semicolon (;) in the first position on the line. The entire line is then considered a comment. Delays may be added using the *Wait* QShell command. Looping and conditional primitives are not supported.

To stop the execution of a batch file prematurely, press *CTRL-D* at the console. It will terminate at the completion of the currently executing command. Any error that occurs within a batch file will cause the batch file to terminate execution.

## SCSI device ID table

The following table may be used to determine the proper hexadecimal value to be used when referencing a SCSI device attached to the system.

SCSI ID	SCSI channel 0	SCSI channel 1	SCSI channel 2	SCSI channel 3
0	0x00	0x10	0x20	0x30
1	0x01	0x11	0x21	0x31
2	0x02	0x12	0x22	0x32
3	0x03	0x13	0x23	0x33
4	0x04	0x14	0x24	0x34
5	0x05	0x15	0x25	0x35
6	0x06	0x16	0x26	0x36
7	RESERVED	RESERVED	RESERVED	RESERVED
8	0x08	0x18	0x28	0x38
9	0x09	0x19	0x29	0x39
10	0x0a	0x1a	0x2a	0x3a
11	0x0b	0x1b	0x2b	0x3b
12	0x0c	0x1c	0x2c	0x3c
13	0x0d	0x1d	0x2d	0x3d
14	0x0e	0x1e	0x2e	0x3e
15	0x0f	0x1f	0x2f	0x3f

## QShell system commands

This section details the available system commands used for command line operations. System commands found in this section are:

- [ACCOUNT](#)
- [ASSIGN](#)
- [BLDCLIP](#)
- [CAPTURE](#)
- [CD](#)
- [CLIPTC](#)
- [CLOCK](#)
- [COMMENT](#)
- [CONFIG](#)
- [CONNCTRL](#)
- [COPY](#)
- [DEL](#)
- [DELTREE](#)
- [DIR](#)
- [DRVMAP](#)
- [DUMP](#)
- [EXIT](#)
- [FS ARRAY](#)
- [FS FORMAT](#)
- [FS MIRRORSETUP](#)
- [GETTC](#)
- [HDDIAG](#)
- [HELP](#)
- [INFO](#)
- [JOG](#)
- [LICENSE](#)
- [LIST](#)
- [LOG](#)
- [LOGIN](#)
- [LOGOUT](#)
- [MIRROR](#)
- [MKDIR](#)
- [MODCLIP](#)
- [MOUNT](#)
- [NET](#)
- [PATH](#)
- [PLAY](#)
- [PLAYSCRIPT](#)
- [POSTDUB](#)
- [PRINT](#)
- [PULLAUD](#)
- [PWD](#)
- [QUEUE](#)
- [QQDOWNLOAD](#)
- [QQUPLOAD](#)
- [RECORD](#)
- [RECOVER](#)
- [REINDEX](#)
- [REN](#)
- [RENVOL](#)
- [REPAIR](#)
- [RESET](#)
- [SCSILIST](#)
- [SERCTRL](#)
- [SET](#)
- [SHELL](#)
- [SHOW](#)
- [SHUTDOWN](#)
- [SHUTTLE](#)
- [STAT](#)
- [STOP](#)
- [TAPE](#)
- [TEMP](#)
- [TIMELINE](#)
- [TYPE](#)
- [UNMOUNT](#)
- [VARSPPEED](#)
- [VER](#)
- [VMODE](#)
- [VOLUTIL](#)
- [VTPMERGE](#)
- [WAIT](#)
- [WHATIS](#)
- [WHICH](#)



## ACCOUNT

### Description

Currently there are three security levels (user, technician, administrator), each allowing the operator a different degree of operator freedom. Security levels are honored from all system interfaces (QShell, Telnet, FTP, GUI).

### **USER**

This is the power-on default and lowest user level. This user level is essentially limited to being able to only play content. Logging out of another level leaves one in the User level. The system log is viewable by users in order to be able to report system warnings or errors.

The Play Browser starts in the SCRIPTS directory for User level access.

### Supported Operations

- Browse and select Show Scripts (playlists) (PLAY – Browser menu page)
- Play content, including starting, stopping and pausing play out (PLAY – Control menu page)
- Issue a predetermined GPI (automation) signal to start, pause, or stop playback.
- View system log

### **TECH**

This security level is for advanced users who are entrusted with content management functions including the loading of new content, encryption key management, removing old content and show script (playlist) management.

### Supported Operations

- All *User* functions
- Content Management
- Playlist (show script) Management

### **ADMIN**

This highest level of access or security provides access to all system functions and menus allowing an Administrator to completely manage all aspects of the systems setup including user security.

### Supported Operations

- All *Technician* functions
- System setup and configuration management
- Security management

### Usage

#### **Account LIST**

**Account ADD <account> <password> <privilege>**

**Account DELETE <account>**

**Account ENABLE <account>**

**Account DISABLE <account>**

**Account IMPORT**

**Account SYSTEM ON|OFF**

### Operations

#### **Account LIST**

Displays the list of defined user accounts. Passwords are hidden from users with *user* and *tech* (technician) permissions. Passwords are displayed for *admin* users.

Example:

```
Volume:>account list
ACCOUNT          PASSWORD          PRIVILEGE          STATUS
-----          -
Richard          medial            admin              Okay
*** Security accounts system is disabled ***
```

#### **Account ADD <account> <password> <privilege>**

This operation is used to define a new user account. Account names and passwords are not case sensitive. This is an ADMIN function only.

**<account>** - This parameter is used to define the account name. A maximum of 15 characters may be used.

**<password>** - This required parameter is used to define the account password. This may be any alphanumeric character. The maximum length of the password is 15 characters. To create an account without an assigned password, use double quotes "" in place of the password parameter.

No password example:

```
Volume:>Account add Projection "" user
```

**<privilege>** - This parameter is used to assign the user security level for the account.

Example:

```
Volume:>Account ADD Richard media1 admin
```

#### **IMPORTANT!**

1. The security system only supports a maximum of 16 user accounts.
2. Accounts names may only have a maximum of 15 characters.
3. Passwords may only have a maximum of 15 characters.

#### **Account DELETE <account>**

This operation is used to remove a user account. This is an ADMIN function only.

Example:

```
Volume:>Account DELETE Richard
```

#### **Account ENABLE <account>**

This operation is used to enable an account that has been disabled. This is an ADMIN function only.

Example:

```
Volume:>Account ENABLE Richard
```

#### **Account DISABLE <account>**

This operation is used to disable an existing user account. This is ADMIN function only.

Example:  
Volume:>**Account DISABLE Richard**

### **Account IMPORT**

This operation is used to import a defined list of user accounts stored in the *users* file (located in the /user/security directory). This is an ADMIN only function.

Example:  
Volume:>**Account IMPORT**

### **Account SYSTEM ON|OFF**

This operation is used to turn system security ON or OFF. After you turn on or off system security you must reset the unit in order for the system security to be properly enabled/disabled. This is an ADMIN only function.

Example:  
Volume:>**Account SYSTEM ON**

## **ASSIGN**

### Description

Assigns any path to a name that you specify. This is useful when you frequently access a path that is many levels deep within the directory structure.

### Usage

```
assign [< name >]
assign < name > <path> ADD
assign < name > <remove>
```

To show all path assignments typing the *assign* command without using any parameters.

```
assign
```

### Parameters

#### **<Path>**

Any directory path. For example *Volume:clips/recent/tests* can be assigned a shortcut path of “tests.” by typing:

```
assign tests Volume:clips/recent/tests ADD
```

You may now navigate to the new path assignment from any directory or volume location by simply typing:

```
cd tests:
```

#### **<Remove>**

To remove the “tests” path assignment type the following:

```
assign tests remove
```

**NOTES:**

1. The system does not save path assignments when the unit is powered off.
2. If you wish to create the same path assignments each time the system is started, you may add the appropriate *assign* commands to the *custom.bat* file that is loaded each time the system is powered on.

**BLDCLIP**Description

*Bldclip* is a command utility that allows you to convert the PLAY events (collapsed clips only) listed in a Show List (script) file into a single collapsed clip. This is useful for creating a completed single-clip version of a script or Show List that consists of multiple segments.

**IMPORTANT!**

1. The first clip in the show list (script) determines the number of audio channels for the entire project. If the first clip has two channels of audio, the output clip will too, *even if any other clips in the show list (script) file have more*. If you have clips with a variety of audio channels, you'll need to have the clip with the highest number of audio channels first, or as a workaround you could insert a one from the clip with highest number of audio channels (perhaps a black frame, which would go unnoticed). The result is that the clip created by *Bldclip* will then have the number of channels needed to accommodate all clips in the show list file.
2. If the timecode attributes of the subsequent clips in the show list do not match that of the first clip (i.e. the show list has a mix of drop frame and non-drop frame clips), then the in and out points of those clips *whose drop frame differs from the first clip will be affected*. This behavior will not appear when you play the show list, but it does affect *Bldclip*. You may have to recalculate in and out points after reviewing the output clip, and rerun the *Bldclip* operation.
3. You cannot mix different image standards or formats. All clips must be of the same format (image size, QMF and frame rate). However, as long as the formats match, you *can* mix clips with different SNR settings. The output clip will only display the SNR of the first clip used.
4. You cannot mix clips with different audio properties (i.e. you cannot *Bldclip* a 16-bit clip with a 20-bit clip). However, if the audio properties match, you *can* mix clips with different numbers of audio channels.
5. *Bldclip* does NOT support expanded clips. This command may only be used if the source clips are collapsed media files (interleaved audio and video).
6. Only PLAY events are used to create the single clip version of the Show List. All other QScript events (GPI, Comment, Switch, etc) are ignored.

Usage

**Bldclip <scriptfile> <outclip>**

Parameters

**<scriptfile>**

This parameter is used to specify the source Show List or script file to be created into a single collapsed clip.

**<outclip>**

This parameter is used to name the new clip. You must provide a unique name.

### Example

In this example a collapsed clip will be created from a simple Show List file (script) named Show.

```
;qscript
PLAY /clips/feature/reel1      00:00:00:00    end
PLAY /clips/feature/reel2      00:00:00:00    end
PLAY /clips/feature/reel3      00:00:00:00    end
```

To build a single clip from the above script file you would issue the following command:

```
Volume:scripts>Bldclip Show newclip
```

## CAPTURE

### Description

This command enables you to take a picture or screen capture of the front panel menu system and save the file as a standard bitmap graphic file. This is particularly helpful for creating internal training documentation. If the current working directory is the */clips* directory, the file will be created there unless a specific path is specified. Once the bitmap file has been created, you may use an FTP client to retrieve the file.

**NOTE:** You must specify the file extension (.bmp) when entering in the file name to be created. The file extension is used by computer software applications to identify the file type. If the extension (.bmp) is not present, computer applications will report the file as being an invalid file type. If the file has been transferred to a computer without its file extension, manually add the .bmp file extension and the file should be recognizable.

### Usage

**Capture <file> [POINTER]**

Example:           Volume:>capture sample.bmp

**NOTE:** This screen capture utility is available on all QuVIS video servers. The *Pointer* option only works on those products that have a pointer-based user interface (GUI). The function of the *Pointer* option is to include the pointer-arrow in the saved screen capture bitmap file. If specified, the *Pointer* option is ignored on a QuVIS Ovation, QuVIS Cinema Player and QuVIS Encore.

## CD

Change Directory

User

### Description

Changes the current QShell directory to the path specified. If no path is specified, the current directory will be displayed. The *CD* command is also used to navigate between mounted storage volumes from the command line. To change to a different directory, you must use a colon (:) at the end the volume name. This colon designates the path as a storage volume.

## Usage

**cd [<path>]**

## Parameters

### <Path>

Used to specify the new directory path.

Example:        *Volume:>cd clips*  
New Path:      *Volume:clips>*

Example – Change Volumes:

*Volume:>cd Removable:*

## CLIPTC Clip Timecode

## Description

ClipTC displays correlated frame and timecode information for a specific clip in a variety of ways, depending upon the parameters used.

## Usage

**Cliptc <name> [<value>]**

**Cliptc <name> showmap**

## Parameters

### <name>

Specify the name of the clip to show timecode information. If you do not specify a *value* the command will return the recorded timecode for frame zero or 00:00:00:00.

Example:        *Volume:clips>cliptc sampleclip*  
                 **Frame 0 of clip has timecode 01:00:00:00**

### <value>

Use this parameter to specify either a specific frame number or timecode value. If you specify a frame number, the recorded timecode value will be returned. If you specify a timecode value (LTC) the frame number will be displayed.

Example:        *Volume:clips>cliptc sampleclip 100*  
                 *Frame 100 of clip has timecode 01:00:03:10*  
  
                 *Volume:clips>cliptc sampleclip 01:00:03:10*  
                 *Timecode 01:00:03:10 found at frame 100*

### Showmap

This parameter is used to display the timecode map of the specified clip. This option may be used to show discontinuities in timecode that may occur in some editing environments.

Example:        *Volume:clips>cliptc sampleclip showmap*  
                 *00000 - 04045 from 01:00:00:00 to 01:02:14:25*

**NOTE:** It is possible that a clip may not have a timecode map depending upon how the clip was created.

## CLOCK

### Description

Displays the current system time of the system or may be used to set the real time clock to the date and time specified.

### Usage

**clock show [UTC]**  
**clock set <MM-DD-YYYY> <HH:MM:SS> [UTC]**

### Parameters

#### Show

Used to display the current system (local) time of the system.

Example:           Volume:>*clock show*  
                  Current date/time: Wed 8-18-2004 10:28:49 [local]

#### [<UTC>]

Use this option to display the system's current date/time in Coordinated Universal Time (UTC). You may also use this option when setting the system clock to specify the date/time to be used is UTC. The local time of the system will automatically set according to the current *general,timezone* Environment Variable setting.

Example:           Volume:>*clock show UTC*  
                  Current date/time: Wed 8-18-2004 16:35:08 [UTC]  
                  Current date/time: Wed 8-18-2004 11:35:08 [local]

#### Set

Use this command option to adjust the current time on the system clock.

**NOTE:** You must specify both the date and time when using the *clock set* command.

Example:           Volume:>*clock set 08-18-2004 11:34:59*  
                  Current date/time: Wed 8-18-2004 11:34:59 [local]

UTC Example:

Volume:>*clock set 08-18-2004 16:34:59 UTC*  
Current date/time: Wed 8-18-2004 16:34:59 [UTC]  
Current date/time: Wed 8-18-2004 11:34:59 [local]

#### <MM-DD-YYYY>

This parameter is used to set the current *month-day-year* in the format specified.

#### <HH:MM:SS>

This parameter is used to set the current time (local or UTC) in the format specified. Hours must be in the 24-hour format. If UTC time is used, the *UTC* command option must be specified to correctly set the system time.

## COMMENT

### Description

This command adds a comment to a file. Comments may be useful reminders of the clip's contents. To properly add a multi-word comment to a file, the text of the comment must be enclosed within quotes ("). Comments stored as part of a file are restricted to 31 characters in length, including spaces.

### Usage

**Comment <filename> <comment>**

### Parameters

#### **<filename>**

Used to specify the name (and path if appropriate) of the file you wish to add comments to.

#### **<comment>**

Used to specify the text of the comments you wish to add the specified file.

Example:        `Volume:clips>comment sampleclip "This is a comment"`

**NOTE:** To view the comments of a file use the *DIR* command. Comments are only properties of a single file and do not become duplicated if that file is copied.

## CONFIG

### Description

This command is used to load a custom configuration (config) file from the command line. User config files are stored in the */user/cfg* directory.

### Usage

**Config <file>**

Example 1 – Load a config file from the */user/cfg* directory

```
Volume:user/cfg>config custom.cfg
```

Example 2 – Load a config file from a different working directory

```
Volume:>config /user/cfg/custom.cfg
```



## CONNCTRL Communication control

### Description

This command is used to administer local and remote connections such as an FTP, Telnet or API client session. This command may be used to shutdown a local (QShell or UCI session) or remote connection.

### Usage

**Connctrl stats**

**Connctrl kill <class> <id>**

### Parameters

#### Stats

Displays a summary of all active user connections.

#### Kill

Use this command option (with parameters) to disconnect or shutdown a specified connection. To obtain <class> and <id> use the *Connctrl stats* command first.

#### <Class>

Specify the type of connection to be shutdown. Types include FTP, Telnet, shell, gui, or api connection.

#### <Id>

This parameter is used to specify the connection identification (ID) number of the user connection to be terminated. The connection ID number is listed in the output of the *Connctrl stats* command.

### Example 1 – Listing active user connections

```
Volume:>connctrl stats
```

Connection	Access	ID	Channel	User	From Address
-----	-----	--	-----	----	-----
Shell	FULL	0x15	5	(none)	(local)
GUI	FULL	0x14	0	(none)	(local)
Telnet	FULL	0x16	2165184	(none)	192.168.1.20,1645
FTP	FULL	0xC	0	anonymous	192.168.1.26,2434

Connection – This column lists all current user connections. The Shell and GUI connections are always present and should be not be disconnected unless absolutely necessary.

Access – List the user permission level for each connection.

ID – Display the service ID for each user connection.

Channel - Lists the connection communication channel the user connection is assigned to.

User – Displays the user name assigned to the connection.

From Address – Identifies where the user of the connection initiated the connection. In the case of a network user, the IP address and port number of the connection will be listed.

### Example 2 – Shutting down an FTP connection using the “kill” option

This administrative example will close the current FTP connection allowing a new connection to be established.

```
Volume2:>connctrl kill ftp 0xC  
Kicking off FTP user
```

**NOTE:** This does not shut down the system's FTP server service.

**IMPORTANT!** If user security is enabled, the *Connctrl* command requires supervisor level permissions in order to execute.

## COPY

### Description

This command creates a copy of a file or clip and stores it in the current directory with a unique name or in a different directory. The contents of directories (files only) may also be copied from one location to another using this command by specifying only the source directory name. The original file (or directory) remains in its original location. The current path is honored, so to copy data residing elsewhere on the system, make sure to include the full path on the command line.

During the copy operation, if a filename already exists in the destination directory, the system will ask if you wish to overwrite the file. To overwrite the file, press “y” or “Y”. If you wish to cancel the operation, press “n” or “N”.

**NOTE:** If source is a directory, then all files in that directory will be copied. Source clips may optionally specify an inframe and outframe. Only that section of the clip will be copied. If wildcards are used, the same in and out frame will be used for each matching clip. If a non-clip is specified as the source, the in and out frame will be ignored.

### Usage

```
copy <srcfile> <destfile>  
copy <srcclip> <destclip> [inframe [outframe]]  
copy [-r] <srcfile> <dir>  
copy [-r] <srcclip> <dir> [inframe [outframe]]  
copy [-r] <srcdir> <dir>
```

### Command Options

#### **[inframe [outframe]]**

This set of options allows a subset of a QuVIS media file or clip to be copied. The **inframe** parameter is used to specify the first or starting frame (in absolute frames starting with zero) of the copy operation. The **outframe** parameter is used to specify the last frame to be copied in the operation. If the outframe parameter is omitted the system will use the last frame of the clip file as the outframe.

#### **[-r]**

This option is used to instruct the system to perform a “recursive” copy. All matching clips/files under the source directory will be copied to the destination directory. The tree structure of the source directory will be preserved, recreated at the destination if needed, during the copy operation.

## Parameters

### **<srcfile> <destfile>**

Use this command structure to copy the specified file *<srcfile>* to the same directory but giving it a new name *<destfile>*.

#### Example

```
Volume:>copy samplefile newfile
```

**TIP:** You may use copy a file to another directory and give it a new name using this command structure. To accomplish this, add the entire path in the *<destfile>* parameter.

### **<srcclip> <destclip> [inframe [outframe]]**

Use this command structure to copy the specified clip file *<srcclip>* to the same directory but giving it a new name. If you wish to copy only a portion of the clip, then use the *inframe* option. If you do not specify the *outframe* the system will automatically create a copy from the *inframe* to the end of the clip.

#### **Copy clip to same directory but give it new name**

```
Volume:>copy sampleclip newclip
```

#### **Copy clip to same director with new name but start at frame 1000**

```
Volume:>copy sampleclip newclip 1000
```

#### **Copy clip to same directory with new name starting from frame zero and stopping on frame 1000 (inclusive).**

```
Volume:>copy sampleclip newclip 0 1000
```

### **[-r] <srcfile> <dir>**

Use this command structure to copy a data file from one directory to another but retaining the same file name. The recursive “-r” option may be used to copy all files within the current directory tree that match the specified name pattern. Wildcards are supported with the recursive copy option.

#### **Copy the selected file to a new directory. File name remains the same.**

```
Volume:>copy samplefile /clips
```

#### **Copy all files within the current directory path that matches the name pattern to a different volume.**

```
Volume:>copy -r s* ext_volume:
```

### **[-r] <srcclip> <dir> [inframe [outframe]]**

Use this command structure to copy the specified clip file *<srcclip>* to a new directory but retaining the same file name. If you wish to copy only a portion of the clip, then use the *inframe* option. If you do not specify the *outframe* the system will automatically create a copy from the *inframe* to the end of the clip.

The recursive “-r” option may be used to copy all files within the current directory tree that match the specified name pattern. Wildcards are supported with the recursive copy option.

**Note:** If multiple clip files are selected to be copied using wildcards or the recursive copy option, the specified *inframe* and *outframe* will be applied to all files being copied.

**Copy a clip to a new directory.**

```
Volume:>copy sampleclip /clips
```

**Copy clip to a new directory but do not include the first 1000 frames.**

```
Volume:>copy sampleclip /clips 1000
```

**Copy clip to a new directory starting from frame zero and stopping on frame 1000 (inclusive).**

```
Volume:>copy sampleclip /clips 0 1000
```

**Copy a subset of all clips within the directory (and subdirectories) that match the name pattern using the recursive copy option.**

```
Volume:>copy -r sample* ext_volume:clips/ 0 1000
```

#### **[-r] <srcdir> <dir>**

Use this command structure, to copy all of the files from one directory to another. The recursive “-r” option may be used to copy all files and subdirectories within the current directory tree that match the specified name pattern. Wildcards are supported with the recursive copy option.

Timeline files (.tl) are not copied as part of this operation. In order to copy subdirectories and their contents, the recursive copy option “-r” must be specified.

**Copy the only the files within the specified directory to a new location.**

```
Volume:>copy /clips /project
```

**Copy the specified directory and all of its contents (including subdirectories) to a new location.**

```
Volume:>copy -r /clips ext_volume:
```

### **Wildcard Usage**

Wildcards such as \* (all) or ? (single character), may be used to specify all or certain files to be copied. Wildcards are flexible and may be placed anywhere within the <srcfile/srcclip> parameter to filter what files are selected for the copy operation.

**Copy all files beginning with “hd” to the “clips” directory.**

```
Volume:>copy hd* /clips
```

**DEL** DeleteDescription

This command deletes the specified file or empty directory. To remove a directory and all of its contents (including files and subdirectories) use the *Deltree* command.

Usage

**Del <path/filename> [-options]**

Parameters**<path/filename>**

The parameter is used to specify the file(s) to be deleted. The directory may be specified if the file(s) to be deleted are located outside the current working directory.

Example 1 – delete file in current directory.

```
Volume:>del samplefile
```

Example 2 – delete file not in current directory.

```
Volume:>del /clips/samplefile
```

**[-options]**

Optional command arguments may be added to the *del* command. These options include:

**-q**

Ask about each item to be deleted. If you select multiple files to be delete you will be asked to confirm your desire to delete each selected file.

**-noquery**

Will automatically delete all of the specified files without asking you to confirm your intention.

**Use of wildcards**

Wildcards such as \* (all), may be used to specify all or certain files to be deleted. Wildcards are flexible and may be placed anywhere within the <filename> parameter to filter what files are selected for the delete operation.

Example – delete all files beginning with “hd” to the “clips” directory.

```
Volume:>del hd* /clips
```

## **DELTREE** Delete tree

### Description

This command is used to delete a directory and all of its contents including files and subdirectories.

### Usage

**Deltree <directory>**

### Parameters

**<directory>**

The parameter is used to specify the directory you wish to remove. You may specify a complete path if the directory you wish to delete is not located in the current path.

Example1 – delete a directory and all of its contents (current path)

```
Volume:>deltree clips
```

Example2 – delete a specific directory and all of its contents

```
Volume:>deltree /clips/test
```

## **DIR** Directory

### Description

This command lists the contents of the specified directory, along with details about each file, clip or directory. If no path is specified, the current directory is used.

### Usage

**Dir [<path>] [-options]**

### Parameters

**[<path>]**

This optional parameter is used to instruct the system to list the contents of a specific directory.

Example 1 – list the contents of the current directory

```
Volume:>dir
```

Example 2 – list the contents of a specific directory

```
Volume:>dir /clips
```

### **[-options]**

Optional command arguments may be added to the *dir* command to show extended information or filter the list to only contain certain items. These options include:

- x** Use this to show extended properties or information.
- v** Used to show revision information for system executable files.
- r** Show all subdirectories below the current directory
- f** Display only files

<b>-d</b>	Display only directories
<b>-c</b>	Display only clips
<b>-m</b>	Display media files (includes clips, virtual tapes and timeline files)
<b>-h</b>	Show hidden files
<b>-p</b>	Pause the output once the display window is full (page breaks)
<b>-b</b>	Do not show header and footer information (bare list)
<b>-l</b>	Display full path names for each file listed
<b>-k</b>	Omit file comments
<b>-sn</b>	Sort by name (use <b>-sn-</b> to show in reverse name order)
<b>-ss</b>	Sort by size (use <b>-ss-</b> to show in reverse size order)
<b>-sd</b>	Sort by timestamp (use <b>-sd-</b> to show in reverse timestamp order)
<b>-u</b>	Display unsorted
<b>-all</b>	Same as <b>-r</b> ( <b>-all</b> is a legacy option)
<b>-alt</b>	Same as <b>-x</b> ( <b>-alt</b> is a legacy option)
<b>-rev</b>	Same as <b>-v</b> ( <b>-rev</b> is a legacy option)

Example – show only clips in sorted-by-name order

```
Volume:>dir -c -sn
```

## DRVMAP

### Description

This command is used to manage the list of SCSI device IDs that the unit stored in memory. Certain commands, like the *SCSILIST* command, use this list of stored IDs to quickly scan each SCSI channel for the presence of a device (hard drive or tape drive). This command allows you to manually add or remove a device ID to the list stored in memory. This is useful when you want to add a device ID to the list without having the system automatically remove any IDs that may not be in current use. The *SCSIList fullscan* command will automatically remove from memory, or this list, any ID that is not currently being used when the command is run.

### Usage

```
DrvMap show
DrvMap add <drive>
DrvMap remove <drive>
```

### Parameters

#### DrvMap show

This will display the list of device IDs per channel that is currently being stored in memory.

Example:

```
Volume:>drvmap show
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
Channel 0: -- -- -- -- -- -- -- -- -- -- -- -- -- --
Channel 1: -- -- -- -- -- -- -- -- -- -- -- -- -- --
Channel 2: XX -- -- -- -- -- -- -- -- -- -- -- -- --
Channel 3: XX -- -- -- -- -- -- -- -- -- -- -- -- --
```

#### DrvMap add <drive>

The *add <drive>* argument enables you to safely add a device ID to the stored ID list. This command only affects the scan list of IDs. The system will automatically detect and mount a drive volume once the drive ID has been added to this list. To add a device or drive ID to the list, the ID must be entered using the hexadecimal method.

```

Example:      Volume:>drvmap add 0x21
              Set and saved to non-volatile memory
              Added drive 0x21 to map

              Volume:>drvmap show
                  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
Channel 0:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
Channel 1:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
Channel 2:  XX XX -- -- -- -- -- -- -- -- -- -- -- -- --
Channel 3:  XX -- -- -- -- -- -- -- -- -- -- -- -- -- --

```

**DrvMap remove <drive>**

The *remove <drive>* argument enables you to safely remove a device ID to the stored ID list. If a hard drive is attached to the unit when the ID was removed from this list, the drive and possibly the volume it was a part of will not be detected the next time the unit is powered on. To remove a SCSI ID from the list, the ID must be entered using the hexadecimal method.

```

Example:      Volume:>drvmap remove 0x21
              Set and saved to non-volatile memory
              Removed drive 0x21 from map

              Volume:>drvmap show
                  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
Channel 0:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
Channel 1:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
Channel 2:  XX -- -- -- -- -- -- -- -- -- -- -- -- -- --
Channel 3:  XX -- -- -- -- -- -- -- -- -- -- -- -- -- --

```

## DUMP

Description

This command is used to read and display the properties of a virtual tape (.vtp) or timeline (.tl) file. File properties that are displayed include; video format, snr, timecode rate, virtual timecode, timecode start value and length of the virtual tape.

Usage

```

Dump TL <file>
Dump VTP <file>

```

Parameters**TL <file>**

The *TL <file>* command argument will display the properties of the specified Timeline (.tl) media file. A timeline file is actually a list of individual edits or clips that make up a specified timeline that has been compiled into machine code for fast processing. The timeline properties that are displayed are: frame rate (in field or frames), duration, number of audio channels and a summary of where each edit is stored.

Example:

```

Volume:sample>dump tl auto.tl
Frame rate: 60 fps (DF)
Duration: 00:32:00;00 (115086 elements)
Audio channels: 0
ID:0

**** Video Storage Map ****

```



```
[SM @ 044482C8 - 044482C8]

**** Timecode ****
[TCDB @ 04425AAD]
00000 - 115085 from 00:58:00;02 to 02:01:59;29
```

**VTP <file>**

The *VTP <file>* command argument is used to display the properties of the specified Virtual Tape file (.vtp). A Virtual Tape (or VTP) is an editable file object that emulates a traditional pre-striped video tape (a pre-formatted tape containing video black and timecode). While Virtual Tapes may not be created or edited on a QuVIS Ovation or Cinema Player server, playback of VTPs is certainly supported. The *dump vtp* command will display each individual edit that makes up the virtual tape.

Example:

```
Volume:>dump vtp ff.vtp
Description:
Format: hd2_1920x1080i59.94
SNR: 54 dB
TC Rate: 30 frames/sec (DF, Src)
Timecode used: virtual, starting at 00:58:00;00
Max Duration: 01:04:00;00
ID:0

*** V I D E O   T R A C K ***
0000: BLACK                                V                                000%
--:--:--:--  --:--:--:--  00:57:59;29  02:01:59;29
```

## EXIT

### Description

This command is used to quit and exit the current QShell or telnet QShell connection.

**Warning!** Do not use this command to close the serial (RS-232 port A) QShell connection. If you quit the QShell session on RS-232 port A, located on the front panel of the unit, you will need to restart the in order to restore QShell access to this serial port.

### Usage

#### **Exit**

Example – quitting a Telnet QShell session without closing the Telnet application.

```
Volume:>exit
```

## FS ARRAY

### Description

This command is used to rebuild a drive volume after one of the volume drives has been replaced. Each drive of a volume is assigned a specific position or “slice ID” within the volume. This order is automatically maintained by the file system (even if drives are placed in the wrong drive order) to ensure that media files are written to the volume in the same order every time. If one of the volume drive members is replaced the volume is broken, as the new drive does not contain any information about the volume it’s a member of. The *FS Array* command rewrites the volume “header” information across all drives of a volume including the replacement drive. This allows the volume to be rebuilt without having to reformat the volume.

### Requirements and Notes

1. The replacement drive must have the same SCSI device ID number as the drive it is replacing.
2. The *FS Array* command only rebuilds the volume header information across all member drives. To rebuild the data stored on the volume other commands must be used.
3. Rebuilding a volume structure using this command does NOT guarantee that the data previously stored on the volume is recoverable. If one drive of a two-drive volume is replaced, then 50% of every media file is missing. This makes all media content previously stored on this volume unplayable. If the rebuilt volume is properly mirrored, mirrored media files may be recovered (copied from the mirror volume back to the primary volume) using the *Mirror* command and the *recover* option.
4. All drives of the volume to be “rebuilt” must be in the proper order according to how they were originally created. Each drive of a volume is assigned a “Slice number” when the volume is created. The drive assigned “Slice 1” is the first or primary drive of the volume. The first drive must be on the lowest SCSI channel when this command is issued. The drive assigned as the number Slice 2 must be on the second lowest SCSI channel. If there are more than two drives in the volume, then each “Slice” or drive must be in the proper order with the highest Slice number being located on the highest SCSI channel number.

The “Slice” number of a drive may be determined by issuing the *List drives verbose* command. The *List volumes verbose* command can also be used to determine the SCSI ID of each drive that is assigned to a volume.

**NOTE:** This lowest SCSI channel number does not necessarily mean that the drive with a Slice number of 1 has to be on SCSI channel 0. It simply means that each drive must in order according to their Slice number with Slice 1 being connected to the lowest SCSI channel. The drive with Slice 1 of a two-drive volume may actually be connected to SCSI channel 2 with the second drive or Slice 2 being connected on SCSI channel 3.

**WARNING!** Improper usage of this command may compromise volume data integrity and system performance. Use this command with caution.

### Usage

**FS array** <drive> <width> [<mirrors>]

## Parameters

### <drive>

This parameter is used to specify the SCSI ID, in hexadecimal format, of the first drive of the QFS volume to be rebuilt. Use the [SCSI device ID table](#) to determine the hexadecimal format for a particular SCSI ID.

### <width>

This option is used to specify the number of drives to be formatted as a single volume. All drives must be attached to different SCSI channels and must have the same device ID.

Example – This example will rebuild the volume information on the two-drive volume where the SCSI ID of the first drive of the volume is 0x23. The system assumes that the replacement drive or second drive of the volume will be located at 0x33.

```
Volume:>fs array 0x23 width 2
```

### [<mirrors>]

This optional parameter is used to specify the number of mirror volumes that are assigned to the volume being rebuilt. This parameter is only needed if converting an old style QFS volume to the new format. Do not use this option unless specifically instructed to by QuVIS Support personnel.

## FS FORMAT

### Description

This command is used to format hard drives and to create a drive volume using the QuVIS File System (QFS). Only QFS-formatted hard drives may be used by any of the QuVIS video server products. Multiple drives may be formatted at the same time to create a multi-drive QFS volume. A QFS volume must contain a minimum of two drives if real-time playback of MDR 60 (maximum data rate of 60MB/sec) material is to be supported.

All QuVIS video servers contain four independent LVD SCSI channels. There are several rules that apply when formatting a QFS volume. They are as follows:

1. All hard drives (up to four) that are to be formatted as a single QFS volume must be assigned with the same SCSI device ID. Device ID may only be manually set on external or peripheral bay devices. Device ID and channel assignment for internal media drives (QuVIS Ovation/Cinema Player/Encore/Acuity) is determined by placement of the drive in a specific internal media bay.
2. Each hard drive must be placed on a separate consecutive SCSI channel. This ensures that there is not multiple drive IDs on the same channel, which would cause integrity problems for the affected SCSI channel.
3. The only two required parameters for the *FS Format* command are <drive> and <name>. However, the default number (width) of drives used by the *FS Format* command is four (4). If the actual number of hard drives being formatted as a single volume is less than four, you must use the *WIDTH* parameter to specify the correct number of drives that will make up the volume. QuVIS recommends that you always use the *width* parameter to avoid unexpected results.
4. Do NOT use spaces when specifying the name of a new volume.

- 5.** If any of the drives to be formatted have previously been used in another drive volume, you may need to temporarily remove that device ID or Ids from the device list stored in memory. The QuVIS File System (on all products) periodically scans each SCSI channel for devices. If a drive that contains any volume header information, the system will automatically mount that drive as a volume. If that happens this will cause problems while attempting to format a new volume.

To remove a device ID from the list stored in memory use the *Drvmap remove* command. Removing the device ID from this list should not cause any problems when formatting a volume, provided that all of the attached SCSI drives are set to automatically spin up.

- 6.** Be very cautious when using this command. Pay attention to all system messages as they may prevent you from inadvertently losing data.

There are no limitations on how many times a hard drive may be formatted although additional steps may be required to reformat a hard drive.

**WARNING!** This command will erase all existing data on the hard drive(s) being formatted. Once a drive has been formatted previous data may not be recovered. Contact QuVIS Customer Support if you need assistance in formatting a drive volume.

SCSI ID	SCSI channel 0	SCSI channel 1	SCSI channel 2	SCSI channel 3
0	0x00	0x10	0x20	0x30
1	0x01	0x11	0x21	0x31
2	0x02	0x12	0x22	0x32
3	0x03	0x13	0x23	0x33
4	0x04	0x14	0x24	0x34
5	0x05	0x15	0x25	0x35
6	0x06	0x16	0x26	0x36
7	RESERVED	RESERVED	RESERVED	RESERVED
8	0x08	0x18	0x28	0x38
9	0x09	0x19	0x29	0x39
10	0x0a	0x1a	0x2a	0x3a
11	0x0b	0x1b	0x2b	0x3b
12	0x0c	0x1c	0x2c	0x3c
13	0x0d	0x1d	0x2d	0x3d
14	0x0e	0x1e	0x2e	0x3e
15	0x0f	0x1f	0x2f	0x3f

## Usage

**FS format <drive> <name> [<options>]**

**Options:**

**WIDTH <drives>**

**SLOWMEGS <megabytes>**

**SHORTSTROKE <endblock>**

## Parameters

**<drive>**

This parameter is used to specify the SCSI ID of the first drive in a QFS volume. If multiple drives were being formatted as a single QFS volume, the first drive would be the drive that is connected to the lowest SCSI channel.

Example – Two drives with a device ID of 3, one placed on SCSI channel 0 (0x03) and the other placed on SCSI channel 1 (0x13). The drive on SCSI channel 0 would be used in the *<drive>* parameter as the starting drive. As only two drives are to be formatted, the *width* parameter must be used.

**NOTE:** SCSI channels 0 and 1 are only available from the rear panel any of the 2RU QuVIS video servers. This includes the QuVIS Ovation, QuVIS Cinema Player and the QuVIS Encore. The above example is assuming that the hard drives being formatted are external to the unit.

#### **<name>**

This parameter is used to give a unique label to the new volume. Do not include a colon (:) in the volume name as this will cause unintended volume referencing issues. A general rule of thumb should be to keep volume names short but descriptive.

#### **[<options>]**

There are number of options that may be used with this command. These options allow you to run specific tests on specific drives or drive storage locations. Multiple options may be run concurrently provided that each option is properly specified using the dash (-).

#### **WIDTH <drives>**

This option is used to specify the number of drives to be formatted as a single volume. All drives must be attached to different SCSI channels and must have the same device ID.

Example: `Volume:>fs format 0x03 width 1`

#### **SLOWMEGS <megabytes>**

This option is used to set the amount of space to be used as data file storage on the drive volume. If this option is not specified, then the volume will be created using the default 500MB for the data storage partition. The data partition is specifically used to store data files like audio files. A separate “fast” partition makes up the remainder of the volume and is used to store all media assets including video clips.

Example:

`Volume:>fs format 0x03 width 1 slowmegs 1000`

## **Practical Examples**

### **Formatting a single drive volume**

Formatting a hard drive in the peripheral bay for use as a removable transport drive is a good example of a single drive volume. In this particular case, the front panel removable peripheral hard drive is connected to SCSI channel three (0x3x) with a default SCSI device ID of three or 0x33 in hexadecimal notation. This configuration is the same on all of the small form-factor 2RU QuVIS servers. In this case, we will also create a separate 1GB data partition for data file storage.

#### **Step 1**

Determine the drive number, in hexadecimal notation, to be used when formatting this single drive volume. If you are unsure of the SCSI device ID of the drive you may wish to issue the *SCSIList* command to list all detected devices. Use the SCSI ID table to determine to list the device ID in hexadecimal format.

#### **Step 2**

Format the drive.

Volume:>*fs format 0x33 Removable width 1 slowmegs 1000*

**Step 3**

Before the drive is formatted you will be asked to confirm the format parameters. If you wish to cancel the operation you may do so by pressing the “n” key. To continue formatting the drive volume, press the “y” key.

The formatting process is likely to take several minutes to complete. Larger capacity drives will take longer to format.

**Step 4**

Once the volume has been formatted, the system will ask if you wish to mount the new volume. Press the “y” key to mount the new volume.

## Formatting a two-drive volume

This example will format and mount a two-drive volume.

**Step 1**

Determine the drive number, in hexadecimal notation, of the first drive of the volume. The first drive of a volume is expected to be the drive connected to the lowest SCSI channel. If you have two drives, one connected to channel 0 and one connected to channel 1, use the drive connected to channel 0 as the source of the *<drive>* parameter. In this example, both drives will be assigned a SCSI device ID of 3.

**Step 2**

Format the drive. Remember, the *Width* parameter must be used when formatting a volume containing less than 4 drives.

Volume:>*fs format 0x03 External width 2 slowmegs 1000*

**Step 3**

If the parameters are entered correctly, press the “y” key to continue the format operation.

**Step 4**

Once the volume has been formatted, the system will ask if you wish to mount the new volume. Press the “y” key to mount the new volume.

## FS MIRRORSETUP

### Description

This command is used to create a mirror volume for an existing single-drive or multi-drive volume. The mirror volume is used to ensure uninterrupted playback of mirrored content. If one of the drives of the primary volume experiences any extended read delays or errors, the system will automatically switch, in most cases transparent to the user, to the “mirror” drive to retrieve the needed data.

QuVIS video servers do not use a true RAID level 1 configuration, as the unit does not have a true RAID controller. If content is mirrored, a duplicate copy of the media file is stored on the mirror volume to ensure playback reliability. Media files are automatically copied to the mirror volume any time it is transferred or copied to the primary volume. Media files can be mirrored manually using the *mirror copy* command.

In the event that a primary volume drive has been replaced and the volume rebuilt using the *FS Array* command, mirrored content may be recovered using the *mirror <clipname> recover* command.

### General Requirements

1. In order for content to be mirrored, the primary volume must have a “mirror” volume assigned to it using the *fs mirrorsetup* command.
2. In order for the system to automatically switch to the mirror volume during playback, in the event of a failure on a primary volume drive, the QuVIS Environment Variable *SCSI,Mirror* must be set to true. To verify the value of this EV you may use the following command:

```
Volume:>show scsi,mirror
```

3. If the *SCSI,Mirror* EV value needs to be changed, use the following command:

```
Volume:>set scsi,mirror true
```

4. The device lds of the “mirror” volume must be one greater than the primary volume. For instance, if the device lds of the primary volume are 0x21 and 0x31, then the “mirror” device lds must be 0x22 and 0x32.

### Usage

**FS mirrorsetup <volume name> <number of mirrors>**

### Parameters

#### <volume name>

This parameter is used to specify the volume that will be assigned a mirror volume.

#### <number of mirrors>

This parameter is used to specify the number of mirrors that should be assigned to the primary volume. Typically, the number of mirrors is “1” although more are possible provided that there the proper number of drives are present to create the additional mirrors.

### Example

In this example, we are mirroring the primary media volume (0x21 and 0x22) of a QuVIS Ovation. The primary volume drives are located in the bottom slot locations with 0x21 on the left and 0x31 on the right. The “mirror” drives are located in the top slot positions in the internal media bays.

```
Volume:>fs mirrorsetup Volume 1
```

## GETTC Get timecode

### Description

Use this command to display the current timecode value in the format shown below. A timecode value will be displayed unless no value has been set or no input is present.

### Usage

**Gettc**

### Output Format

LTC in: (none)

LTC out: (none)

Internal: HH:MM:SS:FF

Example:      *Volume:clips>gettc*  
                  *LTC in:    (none)*  
                  *LTC out: 01:00:02:10*  
                  *Internal: 00:00:03;16*

## HDDIAG

### Description

*HdDiag* is a drive diagnostic utility intended to assess the current health of the SCSI hard drives connected to the unit. **This command may take up to several hours to complete depending upon the number of options selected, the size and number of drives being tested.** All of the tests this diagnostic tool uses are non-destructive.

While this test is being run, no other operations should be running. This includes playing content or transferring (copying) content to the machine.

HdDiag may take up to a minute to begin, due to disk preparation, during which time you may not see any visible feedback or activity.

### Usage

**HdDiag [-options]**

### Parameters

#### HdDiag

When invoked from the command line (with no options) HDDiag will immediately perform a series a non-destructive tests (an ECC verification test, a read verify test, and a write verify test) on all hard drives found on all SCSI channels. These tests are performed concurrently on all channels. To perform a single-drive scan, use the *-d* option.

#### [-options]

There are number of options that may be used with this command. These options allow you to run specific tests on specific drives or drive storage locations. Multiple options may be run concurrently provided that each option is properly specified using the dash (-).

<b>-h or -?</b>	Use this option to display command usage and available options.  Example: <i>Volume:&gt;hddiag -?</i>
<b>-v</b>	Do NOT run the ECC verification test.  Example: <i>Volume:&gt;hddiag -v</i>
<b>-r</b>	Do NOT run the byte check read verification test
<b>-w</b>	Do NOT run the byte check write verification test
<b>-l n</b>	Loop the text "n" number of times. Replace "n" with the number of times you wish to run the test(s). Enter zero (0) to loop indefinitely.  Example: <i>Volume:&gt;hddiag -l 4</i>



**-s n**

Use this option to start the test on a specific block or drive LBA location. Using this option with the “-b n” option is an easy way of targeting a specific location or LBA range on a drive.

Example:

```
Volume:>hddiag -d 0x22 -s 100 -b 1000
```

The previous example will scan the drive on channel 2 with SCSI ID 2 only. It will start scanning at LBA block 100 and scan for 1000 blocks. You could just take the -d option out and the program will scan that LBA range on all SCSI devices mounted.

Special attention must be paid to the fact that drive errors recorded in the logging system with list LBA numbers in a hexadecimal format. You can convert this hexadecimal number to decimal before using *HdDiag* or simply just use the HEX value rather than decimal when entering the *HdDiag* command.

Example - Same LBA range from previous example in hex

```
Volume:>hddiag -d 0x22 -s 0x64 -b 0x3E8
```

**-b n**

Use this option to only run this test on specific number of blocks. See the examples listed in the “-s n” option for usage details.

**-q**

Use this option to only scan a sampling of the drive(s). This option will only scan the first and last 12.5% of the drives. Normally, this option is referred to as the “quick” test however, depending upon the size and number of hard drives being tested even the “quick” test may take several hours to complete.

Example – Run the quick test on all attached drives

```
Volume:>hddiag -q
```

**-d <drive> [<drive> <drive> ...]**

This option enables you specify which hard drives will be tested. You must use the hexadecimal method for specifying each drive to be tested.

Example:           Volume:>hddiag -d 0x00 0x10 0x20

## HELP

### Description

This command is used to display a quick summary of QShell's built-in commands. Note that this list does not include software modules (.mod). To display software modules, type "*dir boot:sys/mods*". Software module or mod command usage is detailed in Appendix D.

### Usage

**Help**  
?

### Sample Output

Assign	Info	Queue	Stop
Auth	Jog	QQdownload	Temp
CD	List	QQupload	TimeCode
Clock	Log	Record	Timeline
ClipTC	Login	Recover	Type
Comment	Logout	RecStat	Unmount
Config	Mirror	Ren	VarSpeed
Copy	MkDir	RenVol	Ver
Del	Mount	SerCtrl	Volutil
Dir	Net	Set	Wait
Encrypt	Path	Shell	WhatIs
Exit	Play	Show	Which
GetTC	PlayScript	Shuttle	
?	Print	Stat	
Help	PWD	StopClip	

## INFO

### Description

Displays information about the specified drive (volume), directory, file or clip. This information, where applicable, includes date stamp, size and permission bits. For clips and media files, the duration is also listed.

Drive volumes details include total size, percentage used and amount free.

### Usage

**Info** [**<drive>** | **<dir>** | **<file>**]

Example 1 – Show volume information

```
Volume:>info
Info for volume 'Volume'
Fast Area
  Size:      67.7 GB
  Used:      96.10%
  Free:      2.0 GB
  Largest:   1.1 GB
Slow Area
  Size:      300.0 MB
  Used:      17.8%
  Free:      246.7 MB
  Largest:   28.8 MB
Width of array: 2
```

*Number of mirrors: 0*

#### Example 2 – Show clip or file information

```

Voume:clips>info sampleclip
sampleclip (col clip)                1.9 GB  RWX-   08-18-2004  12:27:12
  [This is a comment]
  Duration: 00:02:15;00
  MaxDataRate: 60
  FrmRate: 29.97 Hz
  Video: 1920 x 1108 Cb,Ylo,Cr,Yhi
    Bits: 12
    Coded: Yes
    WPSNR: 57 dB
    QMF: QMF2
    Encode: QPE
    Fmt: hd2_1920x1080i59.94 (1080i59.94)
  Audio: 8 channels
    Bits: 24
    Freq: 48000 Hz
  MetaData: No
  Encryption:None

```

## JOG

### Description

Jog is available after you play a clip using the “shuttle” option. Jog allows you to jump to any frame in the current clip.

### Usage

**Jog <frame> [Audio]**  
**Jog STOP [Audio]**

### Parameters

**<frame>**

Use this parameter to specify the frame number you wish playback to hold on.

To use Jog, play a clip with the shuttle option:

```
Volume:clips>play sample_clip shuttle
```

While the clip is playing, type the following to jump to frame 6000:

```
Volume:clips>jog 6000
```

Once in this virtual jog mode, you can continue to enter new frame numbers to jog/jump to without having to re-enter the entire line. Use *jog stop* to exit jog mode from the command line.

### **STOP**

Use this command argument to exit jog mode and stop playback.

```
Example:      Volume:>jog stop
```

**NOTE:** Once in shuttle mode, you may use both jog and shuttle commands, so it's possible to cue up a clip to a specific frame (by jogging to the frame) then begin playback from there (using “shuttle 1”).

## LICENSE

### Description

The License command is used to manage the feature licenses on the target server (QuVIS Acuity, QuVIS Encore, QuVIS Ovation and QuVIS Cinema Player). Previously, license information was stored in a text file and placed in the “/user” directory on the primary storage (OS) volume. License information is now stored on a chip inside the chassis making feature licenses independent of the system storage (hard drives). The utility is used to manage the feature licenses for a particular machine.

### Usage

**License LIST**

**License TEST <feature> [<version>]**

**License ADD <feature> <serialnum> <expiredate> <maxversion> <authentication>**

**License DELETE <feature>**

**License IMPORT <license\_file>**

### Parameters

#### **LIST**

Use this command to display the list of installed licenses.

Example: *Volume:>License list*

#### **TEST <feature> [<version>]**

Use this command to determine if a particular feature has been licensed.

#### **ADD <feature> <serialnum> <expiredate> <maxversion> <authentication>**

Use this command to manually add a feature license to the server.

**IMPORTANT!** Only QuVIS may generate the information that makes up a feature license. However, the text string may be provided either verbally or electronically allowing the customer to “add” the license using this template.

#### **DELETE <feature>**

Use this command to remove a feature license from the stored list.

#### **CAUTION!**

Deleting the primary system license will remove the systems ability to play or record content. If a feature license is accidentally removed, contact QuVIS Customer Support for assistance.

#### **IMPORT <license\_file>**

Use this command to import a legacy license file (licenses.txt).

Example: *Volume:>License import /user/licenses.txt*

## LIST

### Description

The List command is used to display all drive volumes currently mounted, all the available formats that are supported for playback as well as a list of all drives currently attached to the system.

### Usage

**List volumes [refresh] [verbose]**

**List formats [refresh]**

**List drives [refresh] [verbose]**

### Parameters

#### Volumes

Use this command argument to list all of the volumes current mounted.

Example: `Volume:>list volumes`

#### Formats

Use this command argument to list all of the video formats supported for playback.

#### Drives

Use this command argument to list all of the drives attached to the system, including hard drives and peripheral drives like a tape drive.

Example: `Volume:>list drives`

#### [refresh]

This parameter is used to instruct the system to update the list of mounted volumes, installed video formats or attached SCSI devices (drives). This is useful when one or more of these items have been added to the system during the current session.

#### [verbose]

Use this parameter to display detailed information about the attached volumes and attached SCSI devices. Extended information will include device ID's, volume assignment and mirror information.

**Example – This example indicates that a single 4-drive volume is mounted or attached to the system.**

```
Volume:>list volumes verbose
Mounted disk volumes:
Volume:                134.58 GB
    00 10 20 30
```

## LOG

### Description

System errors and events are sent to a logging system that keeps track of everything the system does. The LOG command allows you to review the list of new or recent errors as well as add/subtract event types to be logged.

Log is available as both a command and an environment variable. For general users, the environment variable will be most understandable. Experts may wish to explore both, since they can exert a great deal of control over logging through the log command *set*. The log may be detailed or simple, depending upon how it has been configured to capture events. Items in the log have a class (drive error, video failure, temperature, etc.), a severity (comment, warning, error), a timestamp, an English description, and possibly other information.

The most commonly used command/option combination are *log read*, *log read old* and *log stats*.

**Log read** – displays the most recent *unread* events in the log chronologically.

**Log read old** – displays the *entire* log chronologically.

**Log stats** – displays a summary of the entire log categorized by event type.

If the log becomes full, oldest messages are removed to make room for new ones. Only the most recent 100 messages (limited to 40 characters) are kept in the log, and the log is not cleared by a reboot.

### Usage

**Log READ** [<options>]  
**Log CHECK** [<options>]  
**Log STATS** [clear] [<class options>]  
**Log Add** "comment"

### Options

Options may be used to limit the number or types of messages that will be READ or CHECKED. These options act as filters allowing you to view only those messages that meet the specified criteria.

#### **FROM <date>|<time>**

Use this option to only list log messages that occurred at or after the specified date and time. If you do not specify a time all messages that occurred that day will be listed, provided additional filtering is not used.

#### **TO <date>|<time>**

Use this option to filter out those messages that occur after this date. Only those messages up to and including this date will be returned. If the time is not specified then all messages that occurred that day would be returned.

#### **CRITICALS|ERRORS|WARNINGS|INFO**

Use this option to return those log messages with the specified severity level.

#### **OLD**

Use this option to return all messages, including read and unread, in the log.

#### **CODE <value>**

This option may be used to view only those log messages that contain the specified error code.

#### **TERSE**

This option removes information (timestamp, date and message class) from the returned messages in order to provide a summary of each logged message.

**SHORT**

This option is used to remove extended description information that may exist in the message. Most log messages are not affected by this option.

**+class[,class,...]**

This option is used to add a system class to the logging system. By default, all system classes are automatically logged with the exception of the *Stats* and *Comment* classes.

**-class[,class,...]**

This option is used to remove a system class from the logging system. Removing system classes from the log is *not* recommended. This does not impact system performance but may reduce the ability to identify and correct any problems that may occur with the system.

## Parameters

**Log READ <options>**

Use this command argument to list all unread system events currently recorded in the log.

Example 1 – Read unread log messages

```
Volume:>log read
```

Example 2 – Read all log messages

```
Volume:>log read old
```

Example 3 – Read all log messages in specific date range

```
Volume:>log read old from 9/1/04 to 9/20/04
```

**Log CHECK <options>**

This command argument is used to report the number of new events in the log. The details of the new events will not be displayed using this command structure, as it will only return the number of new events.

To report the total events currently recorded in the log, use the following command:

```
Volume:>log check old
```

**Log STATS [clear] [<class options>]**

This command is used to display the number of messages recorded for each event type.

Example 1 – View log statistics

```
Volume:>log stats
```

Example 2 – Clear log statistics

```
Volume:>log stats clear
```

Example 3 – View log statistics for a specific system class

```
Volume:>log stats +play
```

**Log ADD “comment”**

Use this command to add comments to the log.

Example 1 – Add a comment.

```
Volume:>log add "This is a comment"
```

Example 2 – View new comment

```
Volume:>log read
*** Log Contents ***
Log valid from 01-01-1988 -5:-59:-58.73
09-09-2004 11:01:24.84 Comment TrackingInfo 0 This is a comment
```

## LOGIN

**Description**

Login security is disabled by default, which allows anyone to modify its operation freely. However, if login security is turned on, a valid user name and password must be provided in order to operate the unit. Use this command to login into the system.

**Usage**

**Login <user> [<password>]**

Example – Logging into the system.

```
Volume:>login user
```

## LOGOUT

**Description**

This command logs out the current user and restores the current shell to standard user status.

**Usage**

**Logout**

Example:        Volume:>logout



## MIRROR

### Description

If you have opted to use a mirror volume for redundant playback, the mirror command may be used to clone existing media files (including clips) from its source volume to its mirror volume. Media files are automatically mirrored as they are transferred or copied to the source playback volume.

Media files or clips that have been created using QuClips (using the current system as the direct source) may need to be manually mirrored using this command.

If a drive from the mirrored volume is replaced, then all source clips will need to be re-mirrored. Likewise, if a source drive is replaced, media files may be recovered from the mirrored volume using the *Recover* command.

### Usage

**Mirror <clip> COPY [-options]**  
**Mirror <clip> CHECK [-options]**  
**Mirror <clip> RECOVER [-options]**

### Parameters

#### **COPY**

This command argument is used to copy the specified clip(s) to the mirror volume.

Example 1 – Mirror single clip

```
Volume:>mirror sampleclip copy
```

Example 2 – Mirror multiple clips using wildcards

```
Volume:>mirror sample* copy
```

#### **CHECK**

Use this command argument to verify that the mirrored clip is an exact duplicate of the original.

Example 1 – Verify single clip

```
Volume:>mirror sampleclip check
```

Example 2 – Verify multiple clips using wildcards

```
Volume:>mirror sample* check
```

#### **RECOVER**

This command argument is used to restore the original clip using the file on the mirrored volume as the source. If a hard drive on the primary volume is replaced, then all of the clips on that volume are missing the frames that were stored on the replaced drive. This command allows the operator to restore the clips directly from the mirror volume, provided that the original clips were properly mirrored.

Example 1 – Restore a single clip

```
Volume:>mirror sampleclip recover
```

Example 2 – Restore multiple clips using wildcards

```
Volume:>mirror sample* recover
```

**Options**

There is one option that may be used with any of the three command arguments.

**-D**

This option is used to search into subdirectories for the specified file(s).

Example:           Volume:>*mirror sample\* check -d*

## **MkDIR**   Make Directory

Description

This command is used to create a new subdirectory within the current directory.

Usage

**MkDIR <name>**

Example – Create a new directory

Volume:>mkdir new\_directory

## **MODCLIP**   Modify Clip

Description

This command modifies a clip in various user-selectable ways by creating a copy of the clip according to the specified options. This command may also be used to change the form of a clip using the *collapse* or *expand* operations.

Media files or clips that have been created using QuClips (using the current system as the direct source) may need to be manually mirrored using this command.

Usage

**Modclip <operation> <inclip> <outclip> [-options]**

Parameters**<operation>**

The *Modclip* command may be used to perform several different types of operations on a media file or clip. The three basic operations are:

**Copy**

Creates a copy of the clip (all clip types supported). May be used with options that allow you to create a clip that has a shorter duration or different audio settings such as a fewer number of audio channels.

**Collapse**

This operation is used to convert an expanded clip (QMF2 only) or timeline file (.tl) to a single video and audio (A/V) interleaved or collapsed clip.

**Expand** This operation is used to convert a QMF2 collapsed clip to an expanded clip. If the collapsed clip has both audio and video data, the result of the *expand* operation will be the creation of a minimum of two new clips; a video clip (.v) and an audio clip (.a). If the *-t* option is used, then a separate file (.a0x) will be created for each audio channel converted to expanded clip form.

**<inclip>**

This parameter is used to specify the source clip for the selected operation.

**<outclip>**

This parameter is used to specify the name of the media file or clip that will be created as a result of running this command.

**[-options]**

There are many options that may be used to modify the clip structure using this command. These options are as follows:

**-a,-audchans n** Used to set the number of audio channels to be created in the new clip. If this option is not specified the *modclip* command will create the new clip with the same number of audio channels that are contained in the source clip. If new channels are being added that were not in the source clip, they will contain blank (zero-valued) audio data. This option may be used with *copy* and *expand* operations.

Example – Create a new clip with 8 audio channels

```
Volume:>modclip copy sourceclip newclip -a 8
```

**-b,-bits n** The is option is used to modify the audio bit depth of the new clip. If this option is not specified the original setting from the source clip will be used. This option may be used with the *copy* and *expand* operations.

Example – Create the new clip with an audio bit depth of 24

```
Volume:>modclip copy sourceclip newclip -b 24
```

**-q,-samprate n** Use this option to set the audio sample rate of the new clip. If you use this option to change the audio sample rate to a different value found in the source clip, the new clip will be created will no audio data being copied from the source clip. To insert new audio tracks onto the new clip you will need to use the *Postdub* command. If this option is not specified, *modclip* will use the audio sample rate settings of the source clip when creating the new clip. This option may be used with the *copy* and *expand* operations.

**-s,-start n** Use this option to select the video frame to start at, where *n* is any frame number in the clip. This option is useful for creating a subset of the source clip without having to record. The *COPY* command may also be used for this purpose provided that no other clip modification operations are required. You may use the *ClipTC* command to determine the starting frame number if you know the timecode value of the source clip the new clip will start at. This option may be used by all *modclip* operations.

Example – Copy the source clip starting at frame number 100

```
Volume:>modclip copy sourceclip newclip -s 100
```

**-f,-frame n** This option may be used to specify the number of frames to process of the source clip. This option is useful when creating a new clip that is but a subset of the source clip. If you use this option with the `-s` option you can create a new clip that contains only those frame you specify from the original clip. If you only use this option, the new clip will start beginning with frame number zero of the source clip and will only contain the number of frames you specify using this option. This option may be used by *all* operations.

Example – Create a new clip using only the first 1000 frames from the source clip

```
Volume:>modclip copy sourceclip newclip -f 1000
```

**-r,-rate n** This option is used to change the video playback rate in frames per second where *n* is either 24, 25, 30 or 60. Changing the video playback rate will likely cause audio playback problems. If the new clip is meant to have audio, it may need to be remastered at the appropriate frame rate to avoid playback problems. If this option is not specified, the *modclip* command will use the video frame rate setting of the source clip. This option may only be used by the *copy* operation.

Example:

```
Volume:>modclip copy sourceclip newclip -r 23.98
```

**-d,-drop n** This option is used to Timecode rate where *n* is either 0 (non-drop frame) or 1 (drop-frame). If this option is not specified the *modclip* command will use the setting of the source clip. This option may only be used by the *copy* operation.

**-l,-slow n** This option is used to change the video rate from non-drop to drop frame. This option only modifies a value in the header of the file used by the system to determine what video frame rate to use during playback. This does *not* affect output or stored timecode.

**-c** This option is used as a shortcut method of specifying a *collapse* operation.

Example: `Volume:>modclip -c sourceclip newclip`

**-e,-x** Either “-e” or “-x” may be used as a shortcut method for specifying an *expand* operation.

Example: `Volume:>modclip -e sourceclip newclip`

**-m,-audio** This option is used to specify that only expanded audio clips should be created regardless of whether or not there is video in the source clip. This option may only be used with the *expand* operation.

Example: `Volume:>modclip -e sourceclip newclip -m`

**-p,-video** This option is used to specify that a new video clip (.v) should be created during an *expand* operation. This option may only be used with the *expand* operation.

Example: `Volume:>modclip -e sourceclip newclip -p`

- t,-tile <clip>** This option is used to specify the clip to be as filler for video black when collapsing a timeline file. Only the first frame of the clip will be used to replace the black video of a virtual tape. The “filler” clip must have the same clip properties as the timeline file including: SNR, QMF MDR, Format and Frame Rate. This option may only be used with the *collapse* operation.
- i,-tracks** This option is used to indicate that discrete audio channels (a separate audio file for each channel) should be created during an *expand* operation or that separate audio files (discrete) are used as the source of a collapse operation. Discrete audio files must have the same “base name” as the source file being used during the collapse operation. The system will automatically determine the number and order of the channels based upon the base name and file extension. This option may be used for both *expand* and *collapse* operations.

Example: `Volume:>modclip -c expandclip newclip -i`

## MOUNT

### Description

This command is used to mount the drive volume specified. This command is generally not necessary as the system automatically detects the presence of a new volume and mounts it. Note that this is a hexadecimal drive number; as in 0x00 for drive 0 on SCSI channel 0. See the hexadecimal drive chart [SCSI device ID table](#) for proper numbering.

### Usage

**Mount <volume>**  
**Mount drive <drive number>**

Example 1 – mount a volume using the volume name (must include the semi-colon)

```
Volume:>mount external:
```

Example 2 – mount a volume by specifying the drive number (first drive of the volume)

```
Volume:>mount drive 0x34
```

## NET

### Description

This command is used to start system services. Services are special protocol servers that enable network software client to communicate or control the unit. Network services FTP, Telnet, API, QInfo and QUDAC are automatically started during startup.

### Usage

**Net MOUNT FTP**  
**Net MOUNT TELNET**  
**Net MOUNT API**  
**Net MOUNT QINFO**  
**Net MOUNT QUDAC**  
**Net MOUNT HTTP**  
**Net STATS**

### Parameters

#### **MOUNT FTP**

This command is used to manually start the system's network FTP server service. This service is automatically started at startup. If for some reason the FTP server stops (highly unlikely), consider restarting the system before using this command. If an FTP session hangs, then use the *Connctrl* command to drop the hung session.

#### **MOUNT TELNET**

This command is used to manually start the system's network Telnet server. This service is automatically started at startup. If for some reason the Telnet server stops (highly unlikely), consider restarting the system before using this command. If a Telnet session hangs, then use the *Connctrl* command to drop the hung session.

#### **MOUNT API**

This command is used to manually start the system's network Application Program Interface (API) server. This service is automatically started at startup. If for some reason the API server stops (highly unlikely), consider restarting the system before using this command. If an API session hangs, then use the *Connctrl* command to drop the hung session.

#### **MOUNT QINFO**

This command is used to manually start the system's network QInfo server. This service is used for communication between QuVIS video server products. This service enables networked QuVIS video servers (QuBit ST, DS, EL, QuVIS Ovation, QuVIS Cinema Player, QuVIS Encore and QuVIS Acuity) to exchange information for network copy operations. The network service is automatically started at startup.

#### **MOUNT QuDAC**

This command is used to manually start the system's network QuDAC server. The QuDAC server service is used to provide communication between a QuVIS video server and networked PC running the QuDAC (QuVIS Direct Access and Control) software. QuDAC allows an operator to remotely access and control the unit's menu system as if they were standing in front of the machine. QuDAC automatically detects the type of unit it is connected to, using the systems model number, and presents the appropriate interface view.

#### **MOUNT HTTP**

The system has an HTTP web server service that is used to support subtitling between it and a digital projector. The HTTP web server is not automatically started and must be either manually started using

this command or the system must be instructed, using a *custom.bat* file, to start the service when it powers on.

Example 1 – Starting the service from the command line

```
Volume:>net mount http
```

Example 2 – Starting the HTTP server service at startup

To instruct the system to start the HTTP server at power on, the following line must be added to the *custom.bat* file located in the *user/bat* directory on the primary boot volume.

Net mount HTTP

### STATS

This command is used to view general network statistics and current network connections.

Example:        Volume:>net stats

## PATH

### Description

This command controls the building of the system's search path list. If the ADD option is not specified, the path listed becomes the only path in the search path list. Typing the *Path* command without options will return the currently assigned paths. Please note, that any search paths that are added are not maintained between sessions. If you need to retain custom paths between sessions, then add the appropriate path statements to the *custom.bat* file.

### Usage

**Path [<path> ADD]**

Example 1 – View current path assignments

```
Volume:>path
```

Example 2 – Add a new path to the search list. This example adds the "clips" directory to the search path.

```
Volume:>path Volume:clips add
```

Example 3 – Replace the search path. This example changes the search path to the "clips" directory".

```
Volume:>path Volume:clips
```

## PLAY

### Description

This command is used to begin playback of the specified media file. The play command will play all media files. Show list (script) files are not supported by the *Play* command. To play a script of Show List file, use the *Playscript* command.

To stop playback you must issue the *STOP* command or press the STOP button.

### Usage

**Play <name> [<inpoint> <frames>] [<options>]**

### Parameters

#### Inpoint

Use this command argument to start playback at a specific frame. The inpoint must be entered as a frame number, not a timecode value. To determine the frame number from a timecode value, use the *Timecode* command.

#### Frames

Use this command argument to stop playback after a specified number of frames. To convert a timecode value to a frame number, use the *Timecode* command. This command may only be used if the INPOINT argument is used.

#### [Options]

There are a number of options that may be used to control playback of the specified media file. Options may be combined to provide even greater playback control.

Options:

LOOP, QUEUE, ASYNC, SYNC, SHUTTLE, NOAUDIO, STATS,  
HOLD, AUTOMODE, AUTOTRACK, SPEED x/x  
TrigLTC|TrigInt <hh:mm:ss:ff>  
TrigSys <hh:mm:ss>  
SrcTC, AbsTC, AbsTC+<hh:mm:ss:ff>

**Loop** Play the specified clip repeatedly until *CTRL-C* is entered. The command prompt does not return to the shell session until the loop playback has been stopped.

**Queue** The PLAY command finishes when the clip is queued for playback (almost immediately). May not be used in conjunction with *ASYNC* or *SYNC* commands. This is the default behavior of the PLAY command.

Example: `Volume:>play sampleclip queue`

**Async** The command prompt is returned when playback of the specified clip begins. If another play command is currently running, then the QShell command prompt will hold until this play command is started in the queue. May not be used in conjunction with the *Queue* or *Sync* commands.

Example: `Volume:>play sampleclip async`

**Sync** The command prompt is returned once playback of the selected media file has stopped. May not be used in conjunction with the *Queue* or *Async* commands.



Example: `Volume:>play sampleclip sync`

- Shuttle** Use this option place the selected media file under shuttle playback control. In this mode, the *Shuttle* command can be used to change the speed and/or direction of playback at any time. In this playback mode, the media file will remain in playback control, even if one of the endpoints is reached. To exit this playback mode the *Stop* command must be issued.
- Once in shuttle playback mode, you may also use the *Jog* command to immediately jump to a specific frame number.
- The speed specified using the *Shuttle* command are exactly the same as those specified under the *speed* option.
- Noaudio** Disables audio playback even if the clip was originally recorded with audio.
- Stats** Provides console display of statistics during playback, and a summary note when clip playback is stopped. The statistics that are of interest to the operator includes: ERR (the number of dropped frames or other errors), TC (the current timecode), and DAT (the total amount of data storage read to that point).
- Hold** This is a non-global hold-frame option that can be used to instruct the system to play the clip and hold the final frame indefinitely (until *ctrl-c* or *Stop* is received).
- AutoMode** This command option instructs the system to detect the format of the clip and switch to that format (if the detected format is not already loaded) before beginning playback.
- AutoTrack** This option “tracks” the expected frame position if it were playing at exactly 1x. If for some reason a frame is dropped, playback will slow to get back in sync. If the machine were to somehow get behind the “ideal time”, then it would speed up to get back in sync.
- This does not guarantee when the clip starts, but once it does start, it guarantees that it runs at 1x. Combine this with TRIGTC to make sure it starts in sync with external devices.
- Speed x/x** Specifies the playback speed and direction. Negative numbers cause reverse playback, from outpoint to inpoint. Positive numbers cause forward playback from inpoint to outpoint. The number, (x), can be a whole number or a fraction. All numbers are positive unless a minus sign precedes the speed value. Here are some examples:

0	No motion (for use with <i>Shuttle</i> option only, does not output a still)		
1/3	Forward play at 1/3x	-1/2	Reverse play at 1/2x
1	Forward play at 1x	-2	Reverse play at 2x
11/10	Forward play at 1.1x	-5/2	Reverse play at 2.5x
5	Forward play at 5x		

Example 1 – Play clip at 2x forward.

`Volume:>play sampleclip speed 2`

Example 2 – Reverse playback at 1.5 x

`Volume:>play sampleclip speed -3/2`

**NOTES:**

1. Minimum speed in either direction is 1/250 and maximum speed is 1000.
2. If the speed option is used, the clip will play to the end and cannot be shuttled or jogged.

**TrigLTC | TrigInt <hh:mm:ss:ff>**

Selects the incoming Linear Time Code (LTC) value at which to begin playback. When the incoming timecode matches the <hh:mm:ss:ff> specified, playback begins. If this parameter or the next one is not included, playback begins immediately. The *TrigInt* <hh:mm:ss:ff> option may be used to trigger playback based upon the internal “free-running” timecode generator.

**Trigsys <hh:mm:ss>**

Selects the local system time at which to begin playback. When the system’s internal system clock matches the <hh:mm:ss> specified, playback begins.

**SrcTC or AbsTC or AbsTC+<hh:mm:ss:ff>**

Used to specify the type of timecode to be output during playback. *SrcTC* output the timecode that was recorded with the media file otherwise known as “source timecode”. *AbsTC* outputs “absolute” timecode which is timecode that starts at 00:00:00:00. *AbsTC+<hh:mm:ss:ff>* is used to output absolute timecode plus a user defined offset.

If none of these options are specified, the output timecode will be governed by the “timecode,output” Environment Variable setting.

Example 1 – Output recorded timecode

```
Volume:>play sampleclip SrcTC
```

Example 2 – Output Absolute timecode with user offset (starting at 1 hour)

```
Volume:>play sampleclip AbsTC+01:00:00:00
```

## PLAYSCRIPT

### Description

The *Playscript* command is used to play a “Show List” or “QScript” file from the command line. Default behavior is to play the entire script from beginning to end, with the option of previewing five seconds before and after transition points.

### Usage

**Playscript <scriptname> [-options]**

### Parameters

**<scriptname>**

This is the name of the script or Show List to be played. You may enter the full path and the filename if the file exists in a different directory than where the command is being executed.

Example: `Volume:playscript /scripts/showlist`

**NOTE:** The playscript command will play scripts that reference clip or media file that reside on other mounted volumes provided two conditions are met: all clips are referenced in the show list file by their full pathnames (*volume:path/filename*), and the show list is run from the /scripts directory.

## Options

### -p or -preview

This option will play the show list or script in a preview mode. Preview mode is used to review the transition between “play” events. Each play event is played five seconds before a transition and continues playing for five seconds after the next play event. This is useful when review a long show list without having to watch the entire script or show list.

Example:

```
Volume:scripts>playscript samplescript -p
```

## POSTDUB

### Description

This command is used to replace one or more audio channels of a clip using a WAV or AIFF audio file. Dubbing audio file into a clip requires that the clip already have space reserved for audio channels. To view the audio properties of a clip, use the *STAT* command. If the clip does not contain the proper number of audio channels you may need to modify the clip to create or reserve space for the additional channels. You may use the *Modclip* command to create space for additional audio channels.

### Requirements and Notes

1. Both the clip and the audio file must reside on the QuVIS video server preferably in the same directory. If the audio file (.wav or .aif) is not in the same directory you must specify their location using the full path when referencing the clip file or audio file. If the two files are on separate volumes, you will need to include the volume name as part of the path.
2. The default behavior is for four audio channels to be dubbed at a time, starting with the lowest-numbered channel, for the duration of the clip (from frame 1 to the end). If the audio and video do not match in length, the audio may run out early (followed by silence to the end) or be cut off prematurely. *Postdub* does not mix audio channels, it replaces them.
3. *Postdub* dubs up to four channels at a time. Dubbing more than four channels into a clip requires at least two separate postdub operations. A separate WAV or AIFF file should be used for each operation.
4. If you do not specify the file extension of the audio file, the system will automatically detect the audio files type (.wav or .aiff).
5. Channels from the group 0 through 3, group 4 through 7 and group 8 through 11 cannot be dubbed in the same operation. To dub 6 channels of audio in a clip (channels 0 through 5), channels 4 and 5 would have to be dubbed in a separate operation from the first four.
6. There is currently no way to “pick and choose” channels from a multi-channel AIFF or WAVE audio file when dubbing into a clip. The first channel is used, then the next, up to the number of channels specified (up to four channels may be dubbed in each operation). For this reason, you may wish to create an audio file with only the track(s) you want prior to dubbing or keep them all as single track files for easier operation.

### Usage

```
postdub <clip> <audiofile> [-options]
```

**Options:**

**-b, -basechan n** First channel to dub  
**-c, -chans n** Number of channels to dub  
**-d, -delay n** Audio delay (frames)  
**-t, -trim n** Audio delay (samples)  
**-s, -start n** Start frame to overdub  
**-f, -frame n** Number of frames to overdub

**Parameters****Postdub <clip> <audiofile>**

If you issue the command without any options you will be replacing the number of audio channels (up to four channels at a time starting with channel 0) of the source clip, *<clip>*, contained in the *<audiofile>*.

Example: `Volume:>postdub sample newaudio.wav`

**[-options]**

There are a number of options that allow you to control specific aspects of the *postdub* operation. The usage of the options is as follows:

**-b, -basechan n** This value represents the first channel in the clip to dub audio data into. On the command line, use *-b* followed by a number from 0 to 11 (the value for the letter "n"). If no value is specified, the default base is considered to be channel 0.

Example:

`Volume:>postdub sample newaudio.wav -b 4`

**-c, chans n** Use this option to specify the number of channels that will be dubbed (up to 4 in each operation). The default value for the number of channels is 2.

Use this option in conjunction with the *-b* option to specify which audio channels of the clip to replace.

**-d, -delay n** This option is used to specify where to start reading audio samples out of the audio file in units of "video frames" where "n" may be any number between the start and end of the audio file.

For example, *-d 60* would skip the first two seconds of audio in the audio file (assuming 30Hz video). Increase this value to move the audio earlier in time in relation to the video; decrease it to shift the audio track later in time. The default value is zero (0).

Example:

`Volume:>postdub sample newaudio.wav -d 60`

**-t, -trim n** Use this option to specify where to start reading audio samples out of the audio file in units of audio samples. This may be in addition to the *-d* option if a coarse frame inpoint has been specified. This effectively gives "sub-frame" delay control, specified in audio samples. The default is zero (0).

If the sample rate is based on 44100 sampling, then  $n$  may be from 0 to 1470 (the number of audio samples in one video frame). If your sampling is based on 48000, then  $n$  may be from 0 to 1600.

Example:

```
Volume:>postdub sample newaudio.wav -t 1200
```

**-s, -start  $n$**

Use this option to specify the video clip frame number where dubbing should start. The default is 0, the start of the clip. Dubbing less than the entire clip is useful to verify proper sync on a section of a clip before you *postdub* the entire length of the clip.

Example:

```
Volume:>postdub sample newaudio.wav -s 100
```

**-f, -frame  $n$**

Use this option to specify how many frames of the clip should be dubbed, where  $n$  may be any number. The default is to dub from the starting frame to the end of the clip.

Example:

```
Volume:>postdub sample newaudio.wav -s 100 -f 100
```

**Example 1 - Use default settings**

Dubbing two new stereo tracks into an existing clip, using the default settings. This will over-write the preexisting data in tracks 0 and 1 with the new audio data.

```
Volume:>postdub displayclip newstereo
```

**Example 2 – Dubbing 6 new audio tracks**

Dubbing 6 new tracks into a clip for 5.1 cinema mix. Assuming that there are two different AIFF format stereo tracks, with left/right data for front and rear, and two WAV format mono tracks for center and subwoofer, here are the four command needed to *postdub* these audio files into a single clip.

**Step 1 – Dub the first stereo pair into the first two channels (number 0 and 1).**

```
Volume:>postdub feature frontpair.aiff -b 0 -c 2
```

**Step 2 – Dub the center channel into channel 2.**

```
Volume:>postdub feature center.wav -b 2 -c 1
```

**Step 3 – Dub the sub channel into channel 3.**

```
Volume:>postdub feature sub.wav -b 3 -c 1
```

**Step 4 – Dub the second stereo pair into the second two channels (numbered 4 and 5).**

```
Volume:>postdub feature rearpair.aiff -b 4 -c 2
```

*Typical surround channel assignments are:*

Channel 0 – Left  
Channel 1 = Right  
Channel 2 = Center

Channel 3 = Sub  
 Channel 4 = Left surround  
 Channel 5 = Right surround

SDDS, or Sony Dynamic Digital Sound, utilizes an 8-channel surround matrix: Center, Left, Right, Left-Center, Right-Center, Left-surround, Right-surround, and LFE (sub).

## PRINT

### Description

The *Print* command is used to write text the console. When used in a batch file, argument macros are expanded. The text to be output must be surrounded by double-quotes. If no text is specified, a carriage return/line feed (CR/LF) is displayed.

### Usage

**Print <"text">**

Example – output text string from the command line.

```
Volume:>print "This is some text"
This is some text

Volume:>
```

## PULLAUD

### Description

This command is used extract one or more audio tracks from a clip. *Pullaud* will create a .wav or .aiff format audio file in the same directory as the originating clip. Extraction does not remove the audio data from the clip it simply duplicates it in audio file form. Collapsed and expanded media file types are supported.

Audio tracks are extracted in groups or streams. The *Pullaud* command does not allow you to extract audio channels that span across different groups. For instance, if you wish to extract the 8 channels you must do so in two separate operations. Channel groups are as follows:

Channels 0 – 3  
 Channels 4 – 7  
 Channels 8 - 11

### Usage

**pullaud <clip> <audiofile> [-options]**

**Options:**

**-b,-basechan n** First channel to pull  
**-c,-chans n** Number of channels to pull  
**-s,-start n** Start frame to pull  
**-f,-frame n** Number of frames to pull  
**-w,-wav** Output file will be WAV (AIFF default)  
**-l,-legacy** Input clip is legacy 50/60 Hz (rare)

### Parameters

**Pullaud <clip> <audiofile>**

The *<clip>* is the name of the clip containing the audio to be extracted. If you do not specify any options, only the first two audio tracks or channels will be extracted.

If you issue the command without any options you will be replacing the number of audio channels (up to four channels at a time starting with channel 0) of the source clip, *<clip>*, contained in the *<audiofile>*.

Example: `Volume:>postdub sample newaudio.wav`

### **[-options]**

There are a number of options that allow you to control specific aspects of the *pullaud* operation. The usage of the options is as follows:

**-b, -basechan n** This value represents the first channel in the clip to extract audio data from. On the command line, use *-b* followed by a number from 0 to 11 (the value for the letter "n"). If no value is specified, the default base is considered to be channel 0.

Example:

```
Volume:>pullaud sampleclip extract_audio -b 4
```

**-c, chans n** Use this option to specify the number of channels that will be pulled or extracted from the source clip (up to 4 in each operation). The default value for the number of channels is 2.

Example:

```
Volume:>pullaud sampleclip extract_audio -c 4
```

**-s, -start n** Use this option to specify the starting frame, when you wish to extract audio from a specific point that is not the beginning of the clip. The default is to extract from the starting frame (0) to the end of the clip.

Example:

```
Volume:>pullaud sampleclip extract_audio -s 100
```

**-f, -frame n** Use this option to specify how many frames of audio will be extracted from the clip, where *n* may be any valid number of frames. The default is to extract audio from the starting frame to the end of the clip.

Example:

```
Volume:>pullaud sampleclip ext_audio -s 100 -f 100
```

**-w, -wav** The default audio file format for extracted audio is AIFF (.aiff). Use the *-w* option to extract the audio data as a .wav file instead.

Example:

```
Volume:>pullaud sampleclip ext_audio -w
```

### **Example 1 - Use default settings**

Extract both channels of audio from a 2-channel clip, creating an .aiff audio file.

```
Volume:>pullaud sampleclip extract_audio
```

**Example 2 – Extract 4 channels from the second channel group**

Extract all 4 channels that make up the second channel group (channels 4 – 7), creating a .wav audio file.

```
Volume:>pullaud sampleclip extract_audio -b 4 -c 4 -w
```

## PWD Present Working Directory

Description

Displays the current working directory path on the console. Basically an alias for the CD command with no specified path.

Example:        Volume:>path  
                 CD is Volume:

## QUEUE

Description

The List command is used to display all drive volumes currently mounted, all the available formats that are supported for playback as well as a list of all drives currently attached to the system.

Usage**Queue**

```
Example:                    Volume:clips>queue
                             ID  St  Length      Done  Name
                             -0-----
> 11  3      356        56  Volume:Clips/sampleclip
```

## QQDOWNLOAD

Description

This command is used to receive a point-to-point file (media or data file) transfer from a networked QuVIS video server or computer workstation. Downloaded files are stored in the current directory. No arguments or options are needed. Once this command is started the system will “listen” to network broadcast packets waiting for the sending server or PC workstation to establish a transfer session.

To abort the command, press *CTRL-C*.

**IMPORTANT!** QQDownload can only receive a single file per session. Each file of an expanded media clip will need to be transferred separately. This includes each video (.v) and audio (.a or .a0x) media files.

Usage

**QQDownload [name] [FROM <xxx.xxx.xxx.xxx>]**



## Parameters

### [name]

This optional parameter is used to specify a new name for the file being transferred. If a name is not specified the filename will remain the same as the original file.

### [FROM <xxx.xxx.xxx.xxx>]

This optional parameter is used to instruct the system only to accept a file transfer from a specific IP address. If this option is not specified, the system will accept any file transfer request.

### Example 1 – Receive media file (no options specified)

```
Volume:Clips>qqdownload
Waiting for sender...
Downloading...
Contacted by 192.168.1.5
Receiving 'sampleclip' (686 frames)
100.0% rec'd      Empty
```

### Example 2 – Receive data file (all options)

```
Volume:Clips>qqdownload samplefile FROM 192.168.1.5
Waiting for sender...
Downloading...
Contacted by 192.168.1.5
Receiving 'testfile' (renaming to "samplefile") (file)
100.0% rec'd      Empty
```

## F2Q - Receiving a file from a computer workstation

In order to receive a media or data file from a networked computer workstation, the *F2Q* command must be used on the computer to initiate the transfer. The *F2Q.exe* application software is stored on the QuVIS video server (QuVIS Acuity, Ovation, Encore, Cinema Player) in the */user/win32* directory for Microsoft Windows® computers and in the */user/linux* directory for Linux workstations.

The *F2Q* command is similar to the [QQUPLOAD](#) command used on the QuVIS server. The primary difference is that when initiating a data file transfer from a computer workstation, you must specify the file type if the file to be transferred is a data file. The file type does not need to be specified if the file being transferred is a QuVIS media file or clip.

F2Q Command Usage:            F2Q <name> <IPaddress> [SPLIT | FILE]

### Example 1 – Transferring a QuVIS media file (clip) to a QuVIS video server

```
c:\>f2q sampleclip 192.168.1.4
```

### Example 2 – Transferring a data file to a QuVIS video server

```
C:\>f2q datafile 192.168.1.4 FILE
```

## QQUPLOAD

### Description

The *QQUpload* command is used to send a QuVIS media file or data file to another QuVIS video server or a network computer workstation. The target device must be “listening” to the network before the file upload begins. A QuVIS video server (QuBit, QuVIS Acuity, Cinema Player, Ovation, Encore) must be running the *QQDownload* command in order to receive the file. A computer workstation (Win32 or Linux) must be running the *Q2F* software utility in order to receive file sent by a QuVIS video server.

To abort the command, press *CTRL-C*.

**IMPORTANT!** QQUpload can only send a single file (media or data) per session. Each file of an expanded media clip will need to be transferred separately. This includes each video (.v) and audio (.a or .a0x) media files.

**NOTE:** You do not need to specify the file type to send a media or data file to another QuVIS video server or network computer (running Q2F utility).

### Usage

**QQUpload <clip|file> <to IP address> [start xx] [frames xx]**

### Parameters

#### <clip|file>

Specify the name of the media file (clip) or data file to be transferred. If the file to be transferred is located in a different directory, you must include the directory path when specifying the file name.

#### <to IP address>

You must indicate where the network location (IP address) the file will be copied. You can determine the IP address of a QuVIS video server by looking at the SETUP – Info menu page or by issuing the following QShell command: *show network*. To find out the IP address of a Win32 computer you should be able to enter the command *ipconfig* from a DOS or command prompt.

Example:            `Volume:>qqupload sampleclip 192.168.1.6`

#### Sample Output

```
Volume:Clips>qqupload sampleclip 192.168.1.6
Starting task 24
Sending clip 'sampleclip' to IP: 192.168.1.6
100.0% sent      3635 K/sec (inst)      1159 K/sec (avg)
Done
```

#### Start xx

To send a subset of the source media file, use this parameter to indicate the starting frame number. If you do not use the *frames* parameter this command will send all frames starting from the inpoint (start xx) to the end of the clip.

Example 1 – Define the starting frame number

```
Volume:>qqupload sampleclip 192.168.1.6 start 100
```

Example 2 – Define an in and out points using frame numbers

```
Volume:>qqupload sampleclip 192.168.1.6 start 100 frames 1100
```

### Frames xx

Use this parameter to specify the number of frames that will be copied to the specified network destination. If you do not use the *start xx* parameter, then the clip will be copied starting with the first frame and ending when the specified frame count has been reached.

Example:           Volume:>qqupload sampleclip 192.168.1.6 frames 1000

## Q2F – Sending a file to a computer workstation

If transferring a media file from a QuVIS video server (using the QQDownload command) to a computer workstation; the Q2F utility must be run on the computer (from a command prompt) to prepare workstation for the file transfer. Data files may also be received on a computer workstation using this utility or by using an FTP client. The Q2F.exe application software is stored on the QuVIS video server (QuVIS Acuity, Ovation, Encore, Cinema Player) in the */user/win32* directory for Microsoft Windows® computers and in the */user/linux* directory for Linux workstations.

The Q2F command is similar to the [QQDOWNLOAD](#) command used to receive file transfers on a QuVIS video server. The file type does not need to be specified.

Q2F Usage:    Q2F <filename> [SPLITLARGE | SPLIT] [KEEPXFER]  
              Q2F -s [SPLITLARGE | SPLIT] [KEEPXFER]

**<filename>** - You must specify a filename to be given to the file received on the computer workstation. The filename given to the transferred file (using Q2F) is used to identify the file on the local operating system.

**IMPORTANT!** If a media file (clip) is copied to a computer workstation and is given a different filename than its original name, the media file will only retain that filename while it resides on the workstation. If the media file is copied back to a QuVIS video server (using F2Q) the original filename is restored as the file is being written to the video server. You may use the [name] option to rename the file as it is being received using QQDownload command.

Example – A media file named *Sampleclip* is copied from a QuVIS Acuity to a computer where it is given a filename of *projectclip*. The new filename (projectclip) only exists on the computer. If the *projectclip* file is transferred to a QuVIS video server, the original filename (which is actually stored in the header of the file) is restored as the file is being written.

If you change the name of a data file during the transfer process (in either direction), the filename change is permanent.

### Example 1 – Receiving a media or data file on a computer workstation

```
C:\>q2f tpcccontrol.cfg
Waiting for Qubit to begin transfer...
Receiving data from Qubit 192.168.55.23
File name: tpcccontrol.cfg
Writing to file 'tpcccontrol.cfg'
Received 0 frames
```

## RECORD

### Description

This command is used to real time record an input video stream as a QuVIS playable clip. Audio, if present, may be recorded at the same time using this one command. By default, recorded clips are of the collapsed (interleaved audio and video) type.

**IMPORTANT!** The QuVIS Ovation or QuVIS Cinema player does not support this command, as these products are playback only servers.

In order to record clips from the command line, the system must first be configured to record. This includes setting the *video format*, *input source*, *image quality (SNR)*, *QuVIS Media Format (QMF)*, *Maximum Data Rate (MDR)*, *video mode* and *audio settings*.

**NOTE:** The following are default settings for the record command. Unless these defaults are overridden from the command line, they will affect all recording.

- Newly-recorded clips are always saved in the current directory
- Default clip form is *collapsed*
- Metadata will *not* be recorded as part of the clip
- Recording will be *open-ended* (recording will continue until stopped)

### Setting up video

#### 1. Specify the **video format name**.

For example, for drop-frame HD, type:

```
Volume:>hd2_1920x1080i59.94
```

To see the list of video formats, type:

```
Dir boot:sys/fmt
```

#### 2. Specify the **input source**.

For example, for serial HD (HD-SDI) type:

```
Volume:>set video,inputsrc serialhd
```

Available video inputs may include:  
*analog, serialsd, serialhd*

#### 3. Set the **image quality**.

For example, for 57dB SNR type:

```
Volume:>set video,snr 57
```

Possible SNR image quality settings:  
33, 36, 39, 42, 45, 48, 51,  
54, 57, 60, 63, 66, 69, 72, 75

#### 4. Set the **QuVIS Media Format** file type.

For example, a new format clip would be:

```
Volume:>set general,qmf qmf2
```

Available QMF file type choices:  
QMF1 (collapsed, legacy-compatible),  
QMF2 (collapsed|expanded, QSR 2.x  
and 3.x compatible)

#### 5. Set the **Maximum Data Rate (MDR)**.

For example (30 MB/sec):

```
Volume:>set video,mdr 30
```

QMF1 = MDR 30 only  
QMF2 = MDR 30 or 60 MB/sec. MDR60  
requires a minimum of a two-drive  
volume to record.

#### 6. Place system is **Record-Ready** state. Type:

```
Volume:>vmode r
```

### Setting up Audio

**7.** Select the analog or digital audio inputs. To use the digital inputs, type:

```
Volume:>set audio,analogpair false
```

**8.** Select the audio bit depth.

Bit depth choices are: 16, 20 or 24

Example, for 24-bit, type:

```
Volume:>set audio,bits 24
```

**9.** Specify the audio sample rate.

Sample rate choices may be either: 44.1 or 48

Example, for 48kHz, type:

```
Volume:>set audio,samplerate 48
```

**10.** Specify the number of audio channels to use.

Number of channels may be set to: 0, 1, 2, 4, 6, 8, 10 or 12

Example, to use 4 channels, type:

```
Volume:>set audio,channels 4
```

To stop recording, you may press **CTRL-C** at any time.

### Usage

**Record <name> [<frames>] [<options>]**

### Parameters

#### <name>

This command argument is used to name the recorded clip.

#### Frames

Use this command argument to stop recording after a specified number of frames.

#### [Options]

There are a number of options that may be used with the record command. Options may be combined to provide even greater control over the record process.

Options:

AV/A/V, META, ASYNC, SYNC, SFRAME, REPLACE, NOAUDIO, STATS, SPEED x/x

CFORM EXPANDED|COLLAPSED, AFORM TRACKS, PICKCHANS x

TrigLTC|TrigInt <hh:mm:ss:ff>

TrigSys <hh:mm:ss>

#### AV/A/V

Specifies whether to record audio and video (AV – command default), audio only (A) or video only (V). If you wish to record only audio, a valid video input source must still be provided even though video will not be recorded.

#### META

Record metadata (if present in the serial digital stream).

- ASync** The command prompt is returned as soon as recording begins. You can use the *STOP* command to stop recording early. This option may not be used in conjunction with *Sync*.
- Example: `Volume:>record sampleclip async`
- Sync** The command prompt is returned after recording is complete. You can press **CTRL-C** to stop recording early. May not be used in conjunction with *ASync*.
- Example: `Volume:>record sampleclip sync`
- SFRAME** Specifies single-frame recording. Each press of the **REC** (record) transport button causes one more frame to be captured and stored. Press **CTRL-C** to stop/exit recording mode.
- REPLACE** Normally, recording will not execute if the specified clip already exists. This option will cause an existing clip of the same name to be overwritten.
- Example: `Volume:>record sampleclip replace`
- NOAUDIO** This option will cause the system to record only video.
- STATS** This option is used to display statistics during the record operation. The statistics of particular interest include **Err** (the number of dropped frames or other errors that occurred while recording), **TC** (the current timecode), and **Dat** (the total amount of data recorded to that point).
- Example: `Volume:>record sampleclip stats`
- Speed x/x** Normally 1, but can specify positive fractions less than one to do time-lapse recording. Negative speeds and speeds greater than 1 have no meaning and are not supported.
- Example: `Volume:>record sampleclip speed 1/300`
- CFORM EXPANDED|COLLAPSED** Defines the “clip form” of the assets to be recorded. **Expanded** indicates that the clip will be created as distinct media files with a separate video track and some number of audio files. Clips recorded in the QMF2 domain may either be collapsed or expanded. **Collapsed** means that the audio and video image data exist in an interleaved format within the clip file. QMF1 clips are always of the collapsed type.
- Example: `Volume:>record sampleclip Cform expanded`
- AFORM TRACKS** Audio form is used to define the way each channel of audio is stored to disk when the *CFORM* option is set to *Expanded*. If this parameter is specified with *Tracks*, each audio channel recorded will be stored in a discrete audio file notated with a “.axx” file extension with “xx” indicating the channel that file was recorded on (i.e. “.a04”).
- If this parameter is not specified, when *CFORM* is set to *Expanded*, a “clustered” audio file will be created where all audio channel data is stored in a single audio file or “.a” file.

Example:

```
Volume:>record sampleclip Cform expanded Aform Tracks
```

**PICKCHANS X** This option is used to select specific audio channels to record when the audio form (AFORM) is set to *Tracks*. To select the audio channel(s) to record, the system uses a unique 2 step method which requires the channels to be selected in binary and then converted to a hexadecimal number.

Step 1: Select audio channels using the following template:

Audio Channels (8)	xxxx yyzz zzzz
Binary Selection	1010 0100 0000 (channels 1, 3, and 6)

Step 2: Convert the binary number into hexadecimal

In this case, selecting channels (1, 3, and 6), the hexadecimal value would be **A4**.

**NOTE:** Make sure to use "0x" preceding the hex number to properly identify the value as hexadecimal to the system.

Using the above example, selecting channels 1, 3, and 6 would require the following option:

```
Pickchans 0xA4
```

#### **TrigLTC | TrigInt <hh:mm:ss:ff>**

Selects the incoming Linear Time Code (LTC) value at which to begin recording. When the incoming timecode matches the <hh:mm:ss:ff> specified, recording begins. If this parameter or the next one is not included, recording begins immediately. The *TrigInt* <hh:mm:ss:ff> option may be used to trigger recording based upon the internal "free-running" timecode generator.

Example:

```
Volume:>record sampleclip triglts 01:58:00:00
```

#### **Trigsys <hh:mm:ss>**

Selects the local system time, generated by the systems RTC (real-time clock), at which to begin recording. When the internal system clock matches the <hh:mm:ss> specified, recording begins.

Example:

```
Volume:>record sampleclip trigsys 11:59:00:00
```

## RECOVER

### Description

This command attempts to recover deleted data from a drive volume. If, since the file or clip was deleted, you have written new data to the drive volume, it is unlikely you will be able to recover data that was previously deleted. If you accidentally delete a file use this command immediately. The operating system will list all files it finds that may be recovered. You will be asked if you wish to recover each file as it is found.

Press the “Y” key to recover the listed file. If you do not wish to recover the listed file, press the “N” key. If you wish to abort the recover process, enter CTRL-C and press the N key on the keyboard.

### Usage

**Recover** <volume:> [**all**] [**verbose**]

### Parameters

#### <volume:>

Enter the name of the drive volume the deleted file resided on before it was deleted. You must include the colon at the end of the volume name. If you do not use the *All* option, then the command will complete once you have recovered a single file.

Example:        `Volume:>recover Volume:`

Sample Output:

```
Scanning for deleted files...
Clip named 'custom.cfg' cannot be recovered, data has been overwritten
File named 'test.bat' cannot be recovered, data has been overwritten
file named 'sampleclip' (size 45056 bytes) -- recover?
```

#### **All**

Use this option if you wish to recover multiple files.

#### **Verbose**

This option is used to display more technical details about the deleted file.

## REINDEX

### Description

This command is used to rebuild the index of a collapsed or interleaved video and audio clip. The usage of the command should no longer be necessary, as the indexing of a collapsed clip should occur automatically whenever the clip is created, copied/transferred or restored from a tape archive. However, running the *Reindex* command on a collapsed clip will not affect the integrity of the clip file in any way.

### Usage

**ReIndex** <clip>

Example:        `Volume:>reindex sampleclip`



**REN**    RenameDescription

Use this command to rename a file or directory. Renaming a directory changes its name, even if it has files within it. The *Ren* command may also be used to move a file or directory (even if it is not empty) to a new location (on the same drive volume) by specifying a different path.

Usage

**Ren <oldname> <newname>**

Example 1 – Rename a file

```
Volume:>ren sampleclip sampleclip_old
```

Example 2 – Move a file to a new directory location (on the same volume)

```
Volume:>ren sampleclip /clips/sampleclip
```

**RENVOL**    Rename volumeDescription

This command may be used to rename a volume.

**IMPORTANT!** Do not use the colon when specifying the current or new volume name.

**WARNING!** Changing the name of a volume may impact script playback operations if the volume path is specified with the show list or script. Changing the volume name will break any file that uses the old name as a reference.

Usage

**Renvol <name>**

**Renvol <old name> <new name>**

**Renvol drive <drive> <name>**

Parameters

**Renvol <name>**

If the present working directory is currently assigned to the volume you wish to rename, then you may use this command as it will rename only the current volume.

Example:        `Volume:>ren NewVolume`

**Renvol <old name> <new name>**

Use this command to rename a volume that is not the current working volume. You may use the *List Volumes* command to view all of the mounted volumes.

Example:        `Volume:>renvol Ext_vol Ext_vol_new`

**Renvol drive <drive> <name>**

This command allows you to rename a volume by using a drive designator (hexadecimal) instead of a source volume name. This is considered to be an advanced operation.

Example:        `Volume:>renvol drive 0x00 Volume`

## REPAIR

### Description

This command is used to identify and resolve any file system errors found on a specified volume. In the event that the file system has become corrupted, this command can be used to fix or repair most errors. If command does not resolve any file system issues, contact QuVIS Customer Support for assistance.

### Usage

**Repair <volume:> [VERBOSE] [FIX]**

### Parameters

#### **Repair <volume>**

Issuing the command without any options will scan the file system on the specified volume for errors. You must use the colon (:) when specifying the volume to be scanned.

```
Example:          Volume:>repair Volume:
                  >>> Scan for bad links...
                  >>> Scan for duplicate nodes...
                  >>> Scan for wasted space...
                  Found 0 error(s) on volume
```

#### **[VERBOSE]**

This optional command parameter will display additional technical information during the scan. QuVIS Support personnel may request issuing this command with this option if they need additional information to troubleshoot a problem.

```
Example:          Volume:>repair Volume: verbose
```

#### **[FIX]**

This command option is used to instruct the system to attempt to automatically fix any errors that may be detected. Once you issue this command, you must answer the prompt by pressing the Y key if you want to repair any detected errors. If you wish to cancel the repair operation, press the N key. If you use this command option to repair any detected errors, you should run the command a second time to ensure all errors have been resolved.

```
Example:          Volume:>repair Volume: fix
                  Do repair on disk volume Beauty:/ ? (Y/N)
                  >>> Scan for bad links...
                  >>> Scan for duplicate nodes...
                  >>> Scan for wasted space...
                  Found 0 error(s) on volume
```

## RESET

### Description

Use this command to perform a safe shutdown and restart of the system from the command line. If there are any open communication connections (minus the QShell session you may be using), the system will not reset.

You may force a restart using the *Anyway* option, but you must take the precaution of ensuring that no critical function is currently being performed. You may use the *Connctrl stats* command to view active remote connections.

If someone else is using the unit, please check with them before initiating the reset sequence.

### Usage

#### Reset [Anyway]

Example:            `Volume:>reset`

## SCSILIST

### Description

This command will return a list of devices attached to the SCSI data bus of the system. The system software maintains a list of attached SCSI device IDs. Periodically, this list may need to be updated if you wish to perform an operation that involves a new SCSI device ID. This command will query each SCSI channel for devices and return the list of devices it finds.

### Usage

Usage: **SCSIlst** [**<channel>**] [**<unit>**] [**VERBOSE**] [**CURRENT**] [**FULLSCAN**]

### Parameters

#### SCSIlst

Use this command without any parameters to view the current list of SCSI devices. The system will only scan for devices with ID's that are currently stored in memory.

Example:

```
Volume:>scsilist
Unit Type      Vendor      Product      Rev  Misc
-----
0/0  Disk      SEAGATE     ST336706LW   010A  Wide Sync
1/0  Disk      SEAGATE     ST336706LW   010A  Wide Sync
1/4  Disk      SEAGATE     ST336706LW   010A  Wide Sync
```

#### Channel

Use this parameter to only show those devices attached to a specific SCSI channel.

Example:

```
Volume:>scsilist 0
Unit Type      Vendor      Product      Rev  Misc
-----
0/0  Disk      SEAGATE     ST336706LW   010A  Wide Sync
```

**Unit**

This parameter is used in conjunction with the <channel> parameter to list information about a specific device id on a specific channel. In the example below the <unit> value used is 4 indicating to the system to display information about the device connect to SCSI channel 1 with an ID of 4.

Example:

```
Volume:>scsilist 1 4
Unit Type      Vendor      Product      Rev      Misc
-----
1/4  Disk      SEAGATE      ST336706LW   010A     Wide Sync
```

**Verbose**

Use this option to display more detailed information about each detected SCSI device.

Example:

```
Volume:>scsilist 0 verbose
Unit Type      Vendor      Product      Rev      Misc
-----
0/0  Disk      SEAGATE      ST336706LW   010A     Wide Sync
      Size: 71687369 blocks (34.1 GB)
      Serial#: 3FD1GGMC00007302EDE2
      Jumpers: ID=48  Auto-spin=N  Delay=N  Term=N  Write-prot=N
```

**Current**

The *Current* option displays the currently negotiated transfer speed and data width for each drive. Width can be "narrow" (8 bits) or "wide" (16 bits). Speed is shown in MB/sec (or "async" when in asynchronous mode -- which has no defined transfer speed). The negotiated transfer speed is indicated in the *Misc* column.

**Fullscan**

This option will scan all SCSI channels and every possible SCSI ID for attached devices. If new device ID are found these will added to the list of ID maintained in memory by the system software. If a device ID that was previously stored in memory has been removed from the system, then that ID will also be removed from the list stored in memory.

Example:

```
Volume:>scsilist fullscan
```

## SERCTRL Serial Control

Description

This command is used to activate remote serial control. This command sets the communication protocol to be used on the specified serial port and then places the system in the "ready" state for remote serial control. As long as this command is active, the system will be listening on the specified serial port for control communication from a remote control device.

Usage

```
SerCtrl <protocol> ON <port>
SerCtrl <protocol> KILL
SerCtrl SHOW
```

## Parameters

### <protocol>

This specifies the serial communication protocol to use while the unit is under remote control.

Available protocols include:

- Odetics
  - Only used on RS-422 serial ports
- LDV8000
  - Only used on RS-232 (port B)
- Sony
  - Also known as BVW or Sony 9-pin
  - Only used RS-422 serial ports
- QShell
  - Used to activate a QShell session on the RS-232 B serial port.

### ON

This command argument is used to initiate remote serial control.

### <port>

Use this parameter to specify the serial port that will be used for remote serial control. Serial port assignment is as follows:

Serial Port	Supported Protocol
232b	QShell, LDV8000
422a	Odetics, Sony
422b	Odetics, Sony

### KILL

This command argument is used to stop remote serial control from the command line.

### SHOW

Use this command argument to display a list of active remote serial control processes that may be active.

Example:            `Volume:>serctrl show`

## Operations

The process for placing the unit under remote control from the command line is slightly different than from the front panel menu system. In order to place the system under remote control from QShell you must also set an Environment Variable from the *Protocol* EV group before executing this command.

### Activating BVW Protocol Serial Control

If the BVW (Sony 9-pin) serial protocol is to be used, you must specify the clip or media file to be controlled by setting the *Protocol,BVWclip* Environment Variable.

`Volume:clips>set protocol,bvwclip Volume:clips/sampleclip`

Example 1 – Starting remote serial control using the Sony protocol

SerCtrl command usage:

**`SerCtrl <protocol> ON <port>`**

`Volume:>serctrl sony ON 422a`

## Example 2 – Stopping remote Sony serial control

SerCtrl command usage:

**SerCtrl <protocol> KILL***Volume:>serctrl sony kill***Activating Odetics Protocol Serial Control**

If the Odetics serial protocol is to be used, you must specify the directory that will be used by the controller to build the list of available clips. To set the Odetics protocol source directory use the following command (replace the last argument with the desired directory path):

*Volume:>set protocol,odeticsdir Volume:clips/*

## Example 1 – Starting remote serial control using the Odetics protocol

SerCtrl command usage:

**SerCtrl <protocol> ON <port>***Volume:>serctrl Odetics ON 422a*

## Example 2 – Stopping remote Odetics serial control

SerCtrl command usage:

**SerCtrl <protocol> KILL***Volume:>serctrl Odetics kill***Activating LDV Protocol Serial Control**

If the LDV8000 serial protocol is to be used, you must specify the clip or media file to be remotely controlled.

*Volume:>set protocol,LDVclip Volume:clips/sampleclip*

## Example 1 – Starting remote serial control using the LDV8000 protocol

SerCtrl command usage:

**SerCtrl <protocol> ON <port>***Volume:>serctrl LDV8000 ON 232b*

## Example 2 – Stopping remote LDV8000 serial control

SerCtrl command usage:

**SerCtrl <protocol> KILL***Volume:>serctrl LDV8000 kill*

## SET

### Description

The *Set* command is used to change the values of Environment Variables (EV). Environment Variables are used to define the current configuration of the system. Individual variables (items) are organized in a collection of related variables called groups. The values of many of the EV are automatically determined and assigned by the system especially when operating from the front panel menu system. Configuring the unit from the front panel is actually modifying Environment Variables, in most cases, via the graphical interface.

Operating the unit from the command line (QShell) may require you to manually set certain Environment Variables in order to achieve the proper configuration for a specific operation.

To review the available Environment Variable groups and items, use the *Show* command to display the list of Environment Variables and groups.

### Usage

```
Set <group,item> <value> [SAVE]
Set <group,item> SAVE
Set <group,item> RESTORE
Set <group,item> ?
Set <group,item>
```

### Parameters

#### <group,item>

This parameter is used to specify the Environment Variable *Group* and Environment Variable *Item* that is to be modified. If you use the *Set <group,item>* command without any options the system will return the currently assigned value.

#### <value>

Use this position to enter the new value to be assigned to the specified Environment Variable.

#### SAVE

This command argument is used to instruct the system to store the new value in Non-Volatile Memory (NVM). Please note that only important system Environment Variables are actually stored in NVM. If you attempt to save an EV that cannot be saved, it will result in an error.

#### RESTORE

This command argument may be used to restore the value of an EV that has been saved to NVM, provided the system has not yet been rebooted.

#### ?

This command argument is used to display help or assignable values of a particular EV item.

### Examples

#### Modify the value of an Environment Variable

```
Volume:>set protocol,odeticsdir Volume:clips/
```

#### Using the SAVE option

```
Volume:>set network,ipaddress 192.168.1.10 SAVE
```

### Displaying the assignable values of an EV item

```
Volume:>set general,daylightsavings ?
Legal values for general,daylightsavings are one of:
    None
    USA
general,daylightsavings = USA
```

## SHELL

### Description

This command is used to open a command line shell, QShell, on the specified serial port. Available ports include: 232b, 422a, and 422b.

To quit the QShell session on the specified port, use the Exit command or restart the unit.

**NOTE:** To use QShell on the RS-422 ports, you will need a 422-to-232 serial cable.

### Usage

**Shell <port>**

## SHOW

### Description

The *Show* command is used to display the settings of Environment Variables. If you know the name of the EV item, like *network,ipaddress* you can enter *show network,ipaddress* to view the currently assigned value.

### Usage

**Show <group,item> | ALL | Groups**

### Parameters

**Show <group,item>**

Use this command to view the current value of the specified EV group and/or item.

Example 1 – Display the value of a single EV item.

```
Volume:>show network,ipaddress
network,ipaddress = 192.168.1.6
```

Example 2 – Display the value of all items within an EV group.

```
save:>show network
NETWORK
IPAddress = 192.168.1.6
SubnetMask = 255.255.255.0
DefaultGateway = 192.168.1.1
EthernetAddress = 00.60.FF.00.22.88
MaxFTPconns = 1
AutoNegotiate = TRUE
Speed = 100
```



```
Duplex = HALF
Link = TRUE
```

**Show All**

Use this command to see the assigned values of ALL Environment Variables.

**NOTE:** The output of this command may exceed the display buffer of the terminal program you are using for your QShell session. You may wish to capture the output as text if you wish to review the entire list.

**Show Groups**

Use this command display a complete list of Environment Variable Groups.

```
Example:      Volume:>show groups
              AUDIO
              DEBUG
              DMA
              GENERAL
              LOG
              MEMORY
              NETWORK
              PROTOCOL
              SCSI
              SECURITY
              SERIAL
              TIMECODE
              VIDEO
              GUI
```

## SHUTDOWN

### Description

Use this command to perform a safe shutdown of the system from the command line. If there are any open connections (minus the QShell session you may be using), the system will not shutdown.

You may force a shutdown using the *Anyway* option, but you must take the precaution of ensuring that no critical function is currently being performed. You may use the *connctrl stats* command to view active remote connections. If someone else is using the unit, please check with them before initiating a shutdown sequence.

### Usage

**Shutdown [Anyway]**

```
Example:      Volume:>shutdown
```

## SHUTTLE

### Description

This command enables you to shuttle through a media file from the command line. From QShell, you must first invoke shuttle function using the *PLAY* command. Once in this virtual shuttle mode, you can enter a new shuttle speed without having to re-enter the entire *PLAY* command.

The maximum speed supported by the shuttle command is 1000. To shuttle in reverse place a minus sign in front of the speed value. Most slow shuttle speeds are supported including 1/4x, 1/2x, 3/4x, etc.

To stop playback use the *Shuttle stop* or *STOP* command.

### Usage

**Shuttle <speed> [Audio]**  
**Shuttle stop [Audio]**

### Parameters

#### **Shuttle <speed>**

Use this parameter to change the shuttle speed in the direction specified.

Example 1 – Shuttle forward at 10x

```
Volume:>shuttle 10
```

Example 2 – Shuttle reverse at 1/2x

```
Volume:>shuttle -1/2x
```

#### **Shuttle stop**

Use this command stop playback and exit shuttle control of the media file.

## STAT

### Description

Displays information about the specified drive (volume), directory, file or clip. **See the INFO command for usage details.**

## STOP

### Description

Use this command to halt the current playback operation. The *Stop* command must be issued for each play event in the *Queue*.

### Usage

**STOP**

Example:       Volume:>stop

## TAPE

### Description

The *Tape* command is used to backup and restore QFS data to and from a SCSI helical-scan tape drive. The format of the data on a QuVIS archive tape is a proprietary format. Since QuVIS employs a highly efficient form of real-time compression when creating clips, there is no further data reduction to be had when archiving these media files to tape. Therefore, it is not necessary to turn on the tape drive's built-in compression option. If it defaults to the *on state*, it is not necessary to change as the *tape* command will set the appropriate settings automatically.

When data is backed up to tape, an archive is created. An archive is essentially a container file that holds media files (clips) and other data that had been selected for backup. There is no special meaning applied to a particular collection of items in an archive, there are merely "a collection of selected data" files under a single filename, like a folder and its content. Unlike a folder, once an archive is written, it is considered "closed" and no further data may be added to it, although data may be read from it. A tape may hold one or more archives, up to the limit of the tape's capacity. Archives cannot be spanned across multiple tapes; therefore it is important to make sure you backup files that can fit in the amount of space available on a tape.

As with other forms of data storage, archive tapes must be formatted before they can be used. A full format operation can take several minutes to complete. For that reason, the ability to re-use a pre-formatted tape by erasing it is also provided. Both the erase and format operations can be performed as standalone operations, or in conjunction with writing the first archive on the tape.

Formatted archive tapes may be used and shared amongst all current QuVIS products including the QuBit product line (QuBit ST, DS and EL) as well as the new QuVIS Acuity, QuVIS Ovation, QuVIS Encore and QuVIS Cinema Player.

**IMPORTANT!** Tape archive operations should only be performed while the system is not actively playing or recording.

**TIP:** To abort or cancel a *Tape* operation, press *CTRL-C*.

### Usage

**Tape <operation> [<options>]**

**Operations:**

**LIST**  
**DIR** <archive(s)>  
**DELETE** <archive(s)>  
**BACKUP** <pattern> <archive name>  
**RESTORE** [<pattern>] <archive name> [@<dest path>]  
**VERIFY** [<pattern>] <archive name> [@<dest path>]  
**ERASE**  
**FORMAT**

**Options:**

**-v,-verify** Verify files after backup  
**-s,-subs** Descend into subdirectories  
**-q,-query** Manually confirm each file  
**-c,-clips** Clips only  
**-f,-files** Files only  
**-y,-playlist** Parse playlists and backup referenced clips  
**-e,-erase** Erase all archives on tape before backup  
**-x,-format** Format tape before backup

**-p,-comp**    **Compress data on backup**  
**-2,-2nd**    **Use 2nd tape drive**  
**-u,-unit n**   **Force use of specific tape drive**  
**-b,-backward**   **Write backward-compatible clip archives**

## Operations

### **LIST**

This operation is used to display the names of the archives on the tape. List reports the uncompressed archive size for each individual archive. At the end of the report you will find the amount of space used on the tape by all archives present. The rest of the reports shows, for each archive listed, the serial number of the QuVIS video server that created the archive, how many files it contains, the total data size, and the date and time the archive was created. For details about the contents of any or all of the archives, use the *DIR* operation.

Example with output:

```
Volume:>tape list
QuVIS tape archive utility
```

Searching for Table of Contents...

Archive	Ver	Ser#	Count	Size	Datestamp
Archive1	2	300103	2	5.3 GB	09-23-2004 18:35:02
Archive2	2	300103	2	4.3 MB	09-23-2004 18:57:03

5.3 GB used, 44.1 GB left (10% full)

### **DIR <archive(s)>**

This operation is used to display the contents (files) of an archive on the tape. The *Name* column displays the name of the file that was archived. The *Type* column indicates whether each item is a data file or clip. For data files, the *Size* column indicates the size of the file, in bytes. For clips, the *size* is indicates the length of the clip or media file (expressed in timecode – HH:MM:SS:FF). The *Status* column indicates whether the item was successfully backed up or not. Other indications here can include “absent” or “failed” (if the item was never backed up or failed during the operation). *Datestamp* is the datestamp of the original file or clip.

Example with output:

```
Volume:>tape dir *
QuVIS tape archive utility
```

Searching for Table of Contents...

\*\*\* Archive 'Archive1'            5.3 GB            Default path: /clips/

Name	Type	Size	Status	Datestamp
sampleclip	Clip	00:05:50:02	Okay	09-22-2004 09:55:51
sample1080i	Clip	00:02:15:00	Okay	09-22-2004 10:42:52

5.3 GB used, 44.1 GB left (10% full)

**DELETE <archive(s)>**

This operation is used to delete one or more archives on the tape. Archives on tape exist back-t-back in a linear fashion, by order they written to the tape. To prevent archives from accidentally being deleted, only allows the latest archive added to the tape to be deleted. If you attempt to delete an archive that is not in the last position, the system will abort the *delete* operation without removing any archives from the tape.

This operation is especially handy for deleting the last archive made. For example, maybe the wrong files were backed, or maybe the archive failed or was aborted mid-backup. With this command you delete this archive and recover space on the tape.

Example:        `Volume:>tape delete Archive3`

**BACKUP <pattern> <archive name>**

This operation is used to write the specified files (data and/or media files) to a named tape archive. The *<pattern>* parameter is actually quite flexible and allows you to specify the name of a single file to archive or you may select multiple files by using wildcard characters. For even further control over the *backup* operation, command options may be used.

Example (with wildcards):

```
Transfer:clips>tape backup *.* Archive4
QuVIS tape archive utility

Found 2 items to backup

Searching for Table of Contents...
Locating free space...

*** Creating archive 'Archive4'...
sample.a
100%
sample.v
100%

Updating Table of Contents...

All files (2) backed up okay
```

**RESTORE [<pattern>] <archive name> [@<dest path>]**

This operation is used to load files from a tape archive on the QuVIS video server. The *Restore* command is quite flexibly as it allows to restore a single file from a single archive or multiple files from multiple archives. The default restore operation will restore the files using their original directory path, on the volume the command was issued from, recreating the target directories if needed. Alternately, the *[@<dest path>]* parameter may be used to override the default and write the files to the specified directory path. The default path of file contained within an archive may be determined by issuing the *tape dir <archivename>* command.

Example 1 – Restore all files from “Archive2” to their default location

```
Volume:>tape restore * Archive2
```

Example 2 – Restore all files from all archives to their default location

```
Volume:>tape restore * *
```

Example 3 – Restore all files from all archives to a specified location

```
Volume:>tape restore * * @Volume:restore/clips/
```

**VERIFY [<pattern>] <archive name> [@<dest path>]**

The *Verify* operation compares the files in an archive to the originals (or copies) on disk, and reports if it finds any discrepancies. This is usually used to verify that files can actually be read back successfully out of an archive without errors, ensuring that a reliable backup has been made.

*Verify* can be done as a standalone operation or as part of a *backup* or *restore* operation (using the *-v* option). As a standalone operation it is more powerful, as a pattern can be specified (as well as the filter options: *-clips*, *-files*, and *-query*) to narrow the verify operation to only a subset of the files in the specified archive.

If [<pattern>] is omitted, "\*" or all files, is assumed. Only one archive is processed. If wildcards are present in the <archive name>, then the first archive that matches the pattern is the one to be verified.

If no [@ <dest path>] is included, the files are all verified against the files at their original location. The inclusion of a @ <dest path> overrides this default, and specifies where to look for files (on the video server) to verify against the files on tape.

Example 1 – Verify archives files to files on disk (original location)

```
Volume:>tape verify * Archive2
```

Example 2 – Verify archives file to file on disk (alternate location)

```
Volume:>tape verify * Archive2 @Removable:clips/
```

**ERASE**

This standalone operation is used to quickly empty a tape of all archives, which is faster than formatting a tape. The erase operation is irreversible, and so a warning is presented which the user must acknowledge that any previous data on the tape will be destroyed.

Erasing a tape may also be done as part of a backup, by using the *-e* option switch on the command line. In such cases the tape is first erased and then the backup operation begins.

```
Example:      Volume:>tape erase
              QuVIS tape archive utility

              This will erase all archives on the tape!  ARE YOU SURE? (n/y)
```

**FORMAT**

This operation prepares a new unformatted tape for use by QuVIS video servers. The format process creates the necessary table of contents and data partitions, as well as other control information that is used to identify the tape. The format operation is irreversible, and so a warning is presented which the user must acknowledge that any previous data on the tape will be destroyed.

Formatting a tape may be done as part of a backup operation, by using the *-x* option switch on the command line. In such cases the tape is first formatted and then the backup operation begins. A brand new tape is unformatted and must be prepared for use.

```
Example:      Volume:>tape format
```

Options**-v,-verify**

This option switch is used with a *Backup* operation to instruct the system to compare the files just archived to tape against the original files on disk. This is used to verify that the backup operation created a reliable archive of the original files.

```
Example:      Volume:>tape backup * Archive1 -v
```

- s,-subs** This option switch is used with a *Backup* operation to retrieve files that match the specified pattern from any subdirectory in the current path.
- Example:        `Volume:>tape backup * Archive -s`
- q,-query** The `-query` option, with a *Backup* or *Restore* operation, to manually confirm each file that will be backed up or restored. This option provides a good method for filtering files that are selected during a backup or restore operation. Each file to be backed up or restored is listed individually prior to the execution of the selected operation.
- File confirmation choices include: “y” = include listed file in operation, “n” = do not included listed file in operation, “A” = include all files from this point forward in the operation, “X” = do not include all files from this point forward in the operation.
- Example:
- ```
Volume:>tape restore * Archive3 -q
QuBit tape archive utility

Searching for Table of Contents...
Found archive 'Archive3'
----> Restore 'sample.a'? (n/y/A/X)
----> Restore 'sample.v'? (n/y/A/X)

Restoring files

>>>sample.a                               Clip 00:00:00:25   Okay   09-
                                   17-2004 17:16:36

All files (1) restored okay
```
- c,-clips** This option switch is used to specify that only clip or media files are to be used for a *Backup* or *Restore* operation.
- Example:        `Volume:>tape backup * Archive1 -c`
- f,-files** This option switch is a file filter that specifies only data files are to be used for a *Backup* or *Restore* operation. Clip files that match the `<pattern>` will be ignored and not processed.
- Example:        `Volume:>tape backup * Archive2 -f`
- y,-playlist** This option switch instructs the *Backup* operation that the file listed in the `<pattern>` parameter is a Show List (or script) file. The *Backup* operation will then parse the Show List file and select all clips or media files that correspond to an active PLAY or STILL event for backup. The show list file will also be backed up as part of the operation. If a media file is referenced by multiple events, the one instance of the file will be stored in the tape archive.
- Example:        `Volume:>tape backup Showlist Archive5 -y`
- e,-erase** This option switch is used to instruct the system to erase all archives on the tape before beginning a *Backup* operation.
- Example:        `Volume:>tape backup * Archive -e`

|                     |                                                                                                                                                                                                                                                                       |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-x,-format</b>   | This option switch is used to instruct the system to format the tape before starting a <i>Backup</i> operation.<br><br>Example: <code>Volume:&gt;tape backup * Archive -x</code>                                                                                      |
| <b>-p,-comp</b>     | This option switch is used to instruct the <i>Backup</i> operation to turn on device compression. If this option is necessary, this option should only be used when archiving data files.                                                                             |
| <b>-2,-2nd</b>      | This option switch is used to instruct system to use the second tape drive if multiple tape drives are connected to the system. The tape drive with the highest device ID or is located on the highest SCSI channel number is considered to be the second tape drive. |
| <b>-u,-unit n</b>   | This option switch is used to instruct the <i>Tape</i> utility to use a specific tape drive for the listed operation. The unit number or device id, “n”, must be specified in hexadecimal format.<br><br>Example: <code>Volume:&gt;tape dir * -u 0x23</code>          |
| <b>-b,-backward</b> | This option is used to create an archive that is backward compatible with the first generation QuVIS video server, the QuBit (1.x).                                                                                                                                   |

## TEMP

### Description

This command is used to display the ambient temperature inside the unit. Temperature sensors are strategically located throughout the unit in order to proactively identify abnormal temperature readings. High temperature events may impact playback performance. High temperature events normally only occur if the airflow into the unit is not properly cooled or maintained.

The main processor board (CPU) within the unit houses the main temperature sensor. If you issue the *Temp* command without any parameters, the temperature reading from the CPU board will be displayed.

### Usage

**Temp [<sensor>]**

### Parameters

#### [<sensor>]

This optional parameter enables you to display the temperature associated with a specific sensor. The available temperature sensors are located on the CPU, VSP, and VIO circuit cards.

Example: `Volume:>temp cpu`

**TIP:** If you wish to display the current temperature for the primary hardware cards (CPU, VSP A and VSP B (if installed), you may issue the following command:

`Volume:>temps`



## TIMELINE

### Description

The *Timeline* command provides several processing options for creating a playable timeline from a VTP file. The resulting timeline file is different than timeline that is automatically generated (auto.tl) or updated by the system when a VTP file is played from the user interface or GUI.

Timeline Update does the same thing, but it's smarter, by only doing the conversion process if the timeline needs updating. In other words, the first time you do an UPDATE to convert a VTP to a timeline, it will do the processing required. Repeat the request and it will finish instantly, since no changes were detected (the timeline is up to date). How does it know if there are changes? First, inside each VTP is an edit tracking identifier. This value gets increased each time the VTP is edited, so it's easy to see if the VTP is an edited version (or the same version) as the one previously converted.

Timeline Import is the same as Update, just without this check (the timeline is ALWAYS created again, even if it was up to date).

Timeline IncrUp is a little more complicated. IncrUp is an "incremental update" option which allows the code to do its processing much more quickly in certain circumstances. In the case of doing a 422 edit, each time an edit is added, the entire VTP used to be converted into a brand new timeline each time. When projects get very large (such as entire shows at a post house), this becomes a big burden. The Incremental Update feature is an optimized import operation that can simply convert the changed portions of the timeline, leaving the existing portions alone. In order for this to work, there are several requirements:

### Usage

**Timeline Import <file> <timeline>**  
**Timeline Update <file> <timeline>**  
**Timeline IncrUp <file> <timeline> <channels> <inpt> <outpt>**

### Parameters

**Timeline Import <file> <timeline>**

The *Timeline import* command takes a VTP (and the assets it refers to) and creates a playable timeline. The <file> parameter is used to name the VTP file that will be used to create the timeline file. The <timeline> file parameter is used to specify the name of the new playable timeline (.tl) file. If the specified timeline file already exists, it will automatically be overwritten.

Example: `Volume:>timeline import project.vtp project.tl`

The resulting timeline file may be saved to any valid location including a different volume. However, if the timeline file is stored on another volume and the volume that contains the VTP assets is removed, the timeline will play but only BLACK video will be outputted.

Example:

`Transfer:>timeline import vtp/project.vtp Volume:remote_project.tl`

**Timeline Update <file> <timeline>**

The *Timeline Update* command is smarter than the *Timeline Import* command as it will only create a new timeline file if additional edits have been added to the VTP that have not yet been included in the specified timeline file. In other words, the first time you do an UPDATE to convert a VTP to a timeline, it will do the processing required. Repeat the request and it will finish instantly, since no changes were

detected (the timeline is up to date). How does it know if there are changes? First, inside each VTP is an edit tracking identifier. This value gets increased each time the VTP is edited, so it's easy to see if the VTP is an edited version (or the same version) as the one previously converted.

Example: `Volume:>timeline update project.vtp project.tl`

### **Timeline IncrUp <file> <timeline> <channels> <inpt> <outpt>**

*Timeline IncrUp* is a little more complicated than the first two command structures. *IncrUp* is an "incremental update" option which allows the code to do its processing much more quickly in certain circumstances. In the case of doing a 422 edit, each time an edit is added, the entire VTP used to be converted into a brand new timeline each time. When projects get very large (such as entire shows at a post house), this becomes a big burden. The Incremental Update feature is an optimized import operation that can simply convert the changed portions of the timeline, leaving the existing portions alone. In order for this to work, there are several requirements:

1. There can only be one new edit since the last timeline update. In other words, if two edits are performed on the VTP, and then an *IncrUp* operation is attempted, it will not be eligible for an incremental update, and so a complete timeline rebuild occurs.
2. The new edit must be described in order to guide the conversion process. The in and outpoints are required, as well as a description of the affected channels.
3. The incremental process basically cuts a hole in the timeline from the inpoint to outpoint, on the channels that are marked as effects. Then, just the portion of the VTP falling into this region (inpoint to outpoint, on effects channels) is processed and added into the timeline.

Notice that requirement 1) means that to avoid complete rebuilds, you should perform an Incremental Update after each single edit. The 422 code, as well as VTP edits from the GUI, both do this.

The *Inpt* and *Outpt* are not timecode, but frame numbers -- and they represent the in and out points relative to the VTP's virtual timecode. The 'channels' value is derived by adding together the following values for each channel that was involved in the edit:

| Channel    | Add This Value                         |
|------------|----------------------------------------|
| Video      | 2147483648 (or 0x80000000 hexadecimal) |
| Audio ch 1 | 2                                      |
| Audio ch 2 | 4                                      |
| Audio ch 3 | 8                                      |
| Audio ch 4 | 16                                     |
| Audio ch 5 | 32                                     |
| Audio ch 6 | 64                                     |

## TYPE

### Description

Use this command to display the contents of a text readable file. This is useful for examining the contents of a batch file, script or show list file and even user config files without having to first transfer the file to a computer.

### Usage

**Type <filename> [PAGE]**

Example:            `Volume:>type /user/cfg/custom.cfg`

### Parameters

#### **[PAGE]**

This optional parameter causes the contents of the file being displayed to pause after twenty-five lines. This is especially helpful when reading the contents of a large text file that may not all fit within the display window at the same time.

Example:            `Volume:>type /user/cfg/custom.cfg PAGE`

To continue or show the next page (25 lines) press any key on the keyboard.

To stop listing the contents of the file, press *CTRL-C*.

## UNMOUNT

### Description

This command is used to manually unmount a drive volume from the unit. The drive volume must be currently mounted and must have no open file (i.e. a media file being played at this moment) for this command to succeed.

Dismounting a drive volume, prepares the drive(s) to be removed from the system without requiring the unit to be turned off. Failure to unmount a drive volume before removing it may result in data corruption.

### Usage

**Unmount <volume>**

**Unmount drive <drivenum>**

### Parameters

#### **Unmount <volume>**

Use this command configuration to unmount the specified volume.

Example:            `Volume:>unmount ext_vol:`

#### **Unmount drive <drivenum>**

Use this command configuration to unmount a drive volume by drive ID. The <drivenum> value must be entered in hexadecimal format. You may use the *List volumes verbose* command to determine the SCSI device ID's that make up a specific volume.

Example:            `Volume:>unmount drive 0x23`

## **VAR SPEED** Variable Speed

### Description

This command is similar to the *Shuttle* command except that when transitioning between playback speed changes this command gradually increases speed while reading every frame providing a smoother transition. In comparison, when the playback speed is adjusted using the *Shuttle* command the system will skip frames to immediately change speeds.

In order to use the *Varspeed* command, playback of a media file must first be initiated using the *play <filename> shuttle* option.

**NOTE:** The maximum playback speed (in either direction) using the *Varspeed* command is variable as it depends upon the content data rate. For most practical applications, the maximum *variable playback speed* is 2x.

### Usage

**Varspeed <speed> [Audio]**  
**Varspeed STOP [Audio]**

### Parameters

**Varspeed <speed>**

Use this parameter to change the playback speed in the direction specified.

Example 1 – Forward at 2x

```
Volume:>varspeed 10
```

Example 2 – Reverse at 1/2x

```
Volume:>varspeed -1/2x
```

**Varspeed STOP**

Use this command stop playback and exit shuttle/varspeed control of the media file.

## **VER** Version

### Description

Use this command to query version information from the system. This command can be used to display the version number of system software and circuit cards. Additional information is also available for advanced diagnostic purposes.

### Usage

**Ver [cards | boards] [boardname] chips] [libs] [releases | RLS] [All]**

### Parameters

**[cards | boards]**

This optional parameter lists the type and version of each hardware circuit card installed.

**[boardname]**

This optional parameter lists the version of a specific hardware card.

**[chips]**

This optional parameter displays information about each programmable chip in the system.

**[libs]**

This optional parameter list all the software libraries installed.

**[Releases | RLS]**

This optional parameter lists details about the current OS release version.

Example:        `Volume:>ver rls`

**[All]**

This optional parameter shows a complete summary of system software and hardware version information.

Example:        `Volume:>ver all`

## VMODE

### Description

This command is used to manually control the operating mode of the internal video/audio subsystem. When the unit is first powered on, it may not yet be ready to play especially from the command line. The *Vmode* command is used to place the system in the appropriate state for the desired operation.

When controlling playback from the command line, there are three general steps.

1. Select the video output format.

Example:        `Volume:>hd2_1920x1080i59.94`

2. Place the system in the playback mode.

Example:        `Volume:>vmode p`

3. Begin playback using the *Play* command.

Example:        `Volume:>play sampleclip`

### Usage

**Vmode [P | R | N | I | O]**

### Parameters

**Vmode P**

This command configuration is used to place the unit in PLAY mode. While in this mode, not actively playing content, the transport status indicator on the front panel “dashboard” will indicate [PLAY].

**Vmode R**

This command is used to place the system in RECORD mode. In order to place the system into RECORD the video format must first be loaded. While in this mode, not actively recording content, the front panel “dashboard” will display [RECORD] while in this mode. Only those QuVIS video servers that have record capability support this command.

**Vmode N**

This command places the unit in a NEUTRAL state. Neutral is used to indicate that the unit is ready to receive instructions or commands. The front panel “dashboard” will display [READY] while in this mode.

**Vmode I**

This mode indicates that the unit is ready to receive a QSDTI transfer. The transport status for this mode is [QSDTI-IN].

**Vmode O**

This mode indicates that the unit has configured its HD-SDI output for a QSDTI transfer. The transport status for this mode is [QSDTI-OUT].

## **VOLUTIL** Volume Utility

### Description

This command provides additional system fault tolerance by allowing the “data” partition of a volume to be duplicated (replicated) across all drive of the volume. The “data” partition is the area on the volume where data files (non-clip) are stored. QuVIS media objects, such as clips, are not protected using this command as all clip data resides in another partition on the volume referred to as the “fast area”. Clips or media files may be archived to digital tape media such as AIT-3 tapes, or can be mirrored to an appropriate mirror volume.

The primary benefit of this command is that in the event a drive failure, the drive can be replaced and volume and mirror information restored to the drive. The data partition, which includes the table of contents on the primary volume drive, can be restored from the duplicated data located on one of the other volume drives. This allows the data area of the volume to be rebuilt without reformatting the entire volume and losing all of the data content. If clips or media files have been properly mirrored, those files may be subsequently restored using the *Mirror recover* command.

**IMPORTANT!** This level of fault tolerance is not automatic and must be invoked manually on each volume to obtain the appropriate benefits. It is possible, in the event of a drive failure, that the duplicated data is out-of-date.

### Usage

```
volutil SYSTEM | -S [volume]  
volutil REBUILD | -R [volume] [DRIVE <drive>]
```

### Operations

**SYSTEM | -S [volume]**

This operation copies the system or data area of the primary volume drive to all drives of the current volume unless the volume name is specified. This is a low-level block copy, so depending upon the size of the data area (slow area) this process may take a while to complete.

Example (2-drive volume):

```
Volume:>volutil -s  
Copying Data Area: 1/1 ...  
100% Complete
```

**REBUILD | -R [volume] [DRIVE <drive>]**

This operation is used to rebuild a drive volume that has had a drive replaced. The replacement drive must be the same size and assigned the same device ID as the drive it is replacing. During the “rebuild”

process volume information is written to the replacement drive and the data area is restored from a copy stored on another drive in the volume.

If this command is issued without specifying a volume, the system will attempt to rebuild the current volume. If the current volume does not need to be rebuilt the operation will abort appropriately.

The *Drive <drive>* argument is required to replace a drive that is not the primary volume drive. If the drive is not specified the system will attempt to rebuild the primary drive (Slice 1) of the volume. If the primary volume drive does not need to be rebuilt, the operation will abort appropriately.

**IMPORTANT!** After a volume has been rebuilt, the system must be rebooted.

Example 1 – Rebuilding a volume by replacing the primary volume drive

```
Volume:>volutil -r
System will now rebuild volume Volume: using drive 0x21
Type 'Y/y' to rebuild or 'N/n' to cancel.
Volume Rebuild
Rebuilding drive 0x00 in Volume Volume: ...
100% Complete
```

Example 2 – Rebuilding a specific volume and specifying the replacement drive

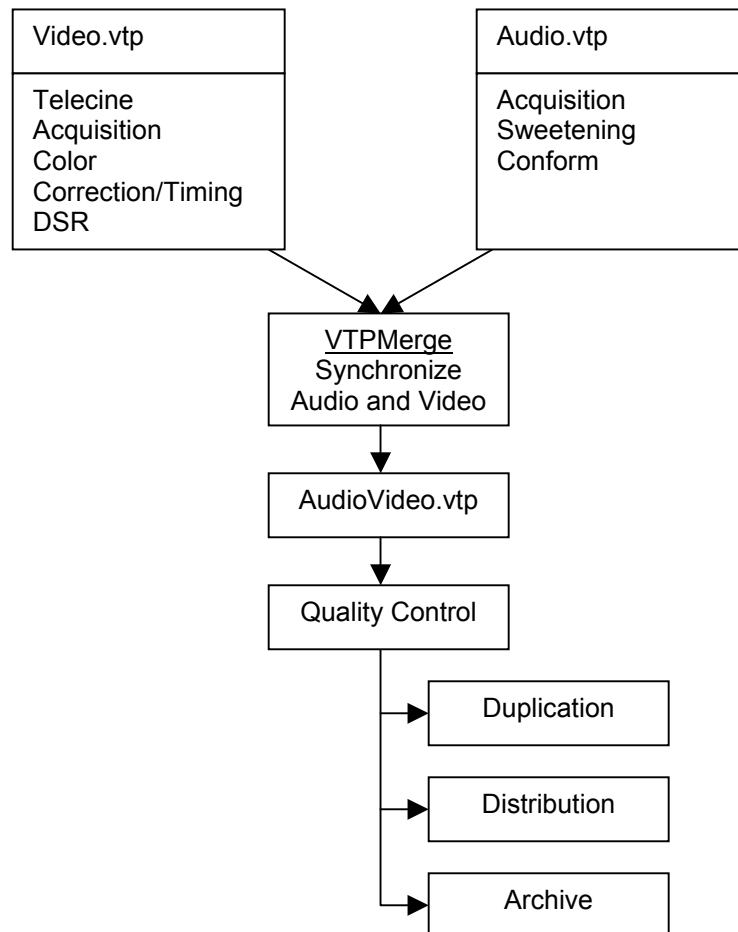
```
Volume:>volutil -r External: 0x32
System will now rebuild volume External: using drive 0x32.
Type 'Y/y' to rebuild or 'N/n' to cancel.
Volume Rebuild
Rebuilding drive 0x32 in Volume External: ...
100% Complete
```

## VTPMERGE

### Description

This command may be used to combine the edits of two Virtual Tape files (.vtp) and create a new VTP file that is a superset of both. The second function this command provides is an audio synchronization mechanism for adjusting the location of all audio assets in relation to the virtual tape timeline. Audio edits may be shifted in either direction, even at a sub-frame level, to achieve better synchronization of video and audio elements.

This basic function of this new utility, called *VTPMerge*, enables one to edit video and audio in separate workflows and then combined later into a single Virtual Tape. For example:



For even greater control of the merge process, you can even adjust the position or placement of the imported audio edits on the “superset” timeline on a frame or sub-frame basis. This allows you to very accurately sync audio to video in preparation for content QC, distribution, duplication or archive.

### Requirements and Notes

1. All source Virtual Tapes must have the same properties (length, video format, SNR, etc.).
2. The timecode (TC) value specified during an AV sync operation is the Virtual Tape TC value.



3. The .xx notation for fractional frame time is only valid for audio TC (this is sub-frame access), and may be omitted if no fractional TC is required for audio.
4. If the destination VTP file already exists, the operation will fail unless the *–replace* option is specified.
5. It is valid (although not recommended) to specify one of the source VTP file names as the destination file name. This would perform a destructive overwrite of the original source VTP file.
6. If multiple source VTP files are used and are physically located on different volumes or directory locations, the removal of one of the volumes or modification of the directory path (such as renaming a parent directory) will cause the new VTP file to be inoperable. For best results, both VTP files should be located on the same volume in the same parent directory.
7. The new VTP file created does not duplicate the existing physical assets of the Virtual Tapes that were used to create it. The new VTP file is a new file that “points” to the existing assets.
8. Deleting the destination VTP will not delete the assets or the .vtp files that were used to create it.

## Usage

**vtpmerge <source VTP file> [<source VTP file>] <dest VTP file> [AVsync vTC aTC] [-options]**

### Options:

**-replace**      Allow overwriting existing VTP file  
**-clean**        Clean VTP(s) of any missing assets

## Parameters

### <source VTP file> [<source vtp file>]

This parameter is used to specify the existing VTP file or files that will be used to create the new superset VTP. The parameter in brackets indicates that multiple source VTP files may be specified. For organizational purposes, it is best if all source VTP files are located in the same parent directory.

### <dest VTP file>

This parameter is used to name the new “superset” VTP file that has been created. To avoid accidentally overwriting existing data, the new VTP file should be given a unique filename.

Example:      `Volume:>VTPMerge video.vtp audio.vtp av_new.vtp`

### [AVsync vTC aTC]

This command option is used to adjust the timing of all audio edits of the source VTP. The timing adjustments move the position of the audio assets in relation to the video timeline in the resulting VTP file. The placement of the video edits on the timeline is not altered. *VTC* is the timeline timecode position where the audio edits will be synchronized or moved to. The *aTC* is the timecode value where the start synchronization point or start frame of audio is currently assigned. An additional timecode position (00:00:00:00.xx) is supported when specifying the *aTC* in order to support sub-frame timing of the audio data. If no fractional time is specified .00 is assumed.

Example – In this example, the audio edits beginning at 1 hour, 8 seconds and 12 frames are moved to a new location on the timeline, 1 hour and 5 seconds.

`Volume:>VTPMerge video.vtp audio.vtp av_new.vtp AVSync 01:00:05:08 01:00:08:12.00`

### [-options]

There are two options that may be used with this command.

**-replace**

This option must be specified if the name of the *<dest VTP file>* already exists and you wish to overwrite it.

Example:

```
Volume:>VTPMerge video.vtp audio2.vtp av_new.vtp -replace
```

**-clean**

This option is used to remove edits from a VTP that point to assets that no longer exist. This enables a corrupted VTP, where the assets may have been deleted, to be updated to only include edits that have physical assets assigned to them.

Example:

```
Volume:>VTPMerge video.vtp video.vtp -replace -clean
```

## WAIT

### Description

This command is used to invoke a delay of the specified number of milliseconds. This command is mainly used in custom batch files to create a pause in processing before the next batch file command.

### Usage

**Wait <ms>**

Example:        Volume:>wait 500

## WHATIS

### Description

Use this command to view any additional details of a specific error code.

### Usage

**Whatis <errorcode>**

### Parameters

**<errorcode>**

This parameter is used to identify the error code you wish to view additional information on. This may be helpful if you encounter an error while performing an operation from the command line.

Example:        Volume:>whatis 0  
0:        Okay

## WHICH

### Description

This command will return the location of the file that can be run from the front panel as a command.

### Usage

**Which <name>**

### Parameters

**<name>**

This parameter is used to specify the name of the software module, batch file or other executable file. Do not include the file extension when specifying the name.

Example:                      Volume:>*which tape*