

# Table Of Contents

Appendices	Appendix A:	Serial Port Troubleshooting
	Appendix B:	Ethernet Troubleshooting
	Appendix C:	Customizing QuBit through Environment Variables
	Appendix D:	Pinout Diagram: Digital Audio Connector
	Appendix E:	Pinout Diagram: Parallel Digital I/O
	Appendix F:	Updating Your QuBit Operating System Software
	Appendix G:	Pinout Diagram: GPI Trigger
	Appendix H:	Shell Commands
	Appendix I:	Exabyte Mammoth Tape Drive
	Appendix J:	Backup/Restore: The Tape Utility
	Appendix K:	How to Create a Play List
	Appendix L:	Customizing QuBit's Startup
	Appendix M:	Mirroring QuBit Hard Drives
	Appendix N:	BVW75 Edit VTR Commands
	Appendix O:	Startup Diagnostics
	Appendix P:	CineGUI Theatrical Interface
	Appendix Q:	Synchronizing Multi-QuBit Playback with Time Code
	Appendix R:	Recording and Playback with Serial/Parallel HD Converters
	Appendix S:	Security Features
	Appendix T:	(none)
	Appendix U:	Creating Uncompressed D-1 Animations
	Appendix V:	LDV 8000 Protocol
	Appendix W:	(none)
	Appendix X:	How to Transfer Files to/from QuBit
	Appendix Y:	(none)
	Appendix Z:	Document History

---

## QuVIS, Inc.

### SUPPORT

800-554-8116 (TOLL FREE VOICE INSIDE US)

785-272-3656 (TOLL FREE VOICE OUTSIDE US)

785-272-3657 (FAX)

support@quvis.com (SUPPORT)

### **Downloading Updates to the QuBit Manuals**

The QuBit Manuals are always being updated with new information, changes, and tips. If you have Internet access, use either a recent version browser or an ftp program and go to:

**[ftp://support.quvis.com/pub/Qubit\\_manual/](ftp://support.quvis.com/pub/Qubit_manual/)**

This directory always contains the latest version of the manual. Each filename contains the date that it was created, so it should be easy to determine if you need to download any newer files.

In addition, at the end of each file there "history" page detailing the changes that were made to each version (so you can tell what has been updated since your previous version).

## Appendix A: Troubleshooting the Serial Connection

In theory many types of devices can share I/O on COM ports, but in real-world operation many computer systems cannot. If your computer cannot establish contact with QuBit, check these items first.

Some computers have add-on devices or installed software that can interfere with COM port availability. If you own a Palm Pilot and use the HotSync Manager software, it's likely you will need to shut it off when you wish to communicate with QuBit. Some IBM ThinkPad laptops with MWave software also use the primary COM port. Having these active and running will almost certainly use the COM port and hinder any communications with QuBit.

Windows users should also check the Device Manager tab (available from the System icon in the Control Panel). Expand the display item for Ports to see if another device is using the port you have selected for QuBit.

You may need to do one or more of the following to solve the problem:

(a) Verify that the COM port settings are correct. They should be 8-N-1, 38400, or:

Data bits:	8
Parity:	None
Stop bits:	1
Baud rate:	38400
Flow Control:	None

(b) Select a different COM port from within the software you are using. Windows systems typically have 2 or 4 COM ports.

(c) Disable the software that's conflicting with the COM port (i.e., in the case of a Palm Pilot, Exit or Quit the HotSync Manager).

(d) Remove the conflicting device altogether (you may need to reinstall it). This may either a software device or a physical hardware device that you need to remove in order to hook up a serial null modem cable to QuBit.

**Important:** Make sure you can restore the system to its previous state of functionality before undertaking system changes of this nature. If not, have your system administrator or staff engineer assist you.



## Appendix B: Troubleshooting the Ethernet connection

If you are unable to establish a network connection to QuBit, check the following situations. We also recommend that you verify the settings on your computer's network control panel (examples for popular operating systems can be found below and on the next page). As long as you make no changes to the control panel gadgets, you can view the settings and exit the panel safely. If you *do* make changes to any panel, Windows will usually request that you shut down and restart the system, which you should do in order for the changes to take effect.

### **QuBit's rear-panel Ethernet lights are not illuminated**

When you connect an Ethernet cable from a known working network you should see several of these lamps illuminate, even if there is no network activity. If they do not light, reboot QuBit. The Ethernet port may be "hung" and unable to connect.

### **Connection is right, but the computer and QuBit do not "see" one another (a).**

QuBit requires a *static IP address*. If the TCP/IP protocol panel on your computer does not show an *assigned* IP address (and there is no gateway that assigns one), you will need to assign an IP address in order to connect to QuBit. To check this, see the examples below for instructions how to verify the current settings and change them if necessary.

### **Connection is right, but the computer and QuBit do not "see" one another (b).**

Check the *Subnet Mask* on your computer. It *must* match the Subnet Mask set up for QuBit. See the examples below to verify the current settings, and change them if necessary.

---

### **Verifying/Changing Network Settings In Windows 95/98 (all versions)**

From the desktop, right-click the **Network Neighborhood** icon and select **Properties**

Select the **TCP/IP** entry for your Network Card

then select **Properties**

and choose the **IP Address** tab

Enter the correct data for **IP Address** \*

and if necessary, the **Subnet Mask** (this value must match the Subnet Mask on QuBit)

and **Default Gateway** \*

\* *If you do not know the values to enter for IP Address or Default Gateway, see your network admin.*

---

### **Verifying/Changing Network Settings In Windows NT 4.0 (SP5 or higher)**

From the desktop, right-click the **Network Neighborhood** icon and select **Properties**

Select the **Protocols** tab

Double-click on the **TCP/IP Protocol** entry

If you do not see an IP address specified, click the radio button marked **Specify an IP Address** then enter the correct data for **IP Address** \*

and, if necessary, the **Subnet Mask** (this value must match the Subnet Mask on QuBit)

and **Default Gateway** \*

*\* If you do not know the values to enter for IP Address or Default Gateway, see your network admin.*

### **Verifying/Changing Network Settings In Windows 2000/Me (all versions)**

From the desktop, right-click the **My Network Places** icon and select **Properties**

Double-click on the icon labeled **Local Area Connection**

In the new window that appears, click on the **Properties** button.

Click on the words **Internet Protocol(TCP/IP)**

Click on the **Properties** button.

If you do not see an IP address specified, click the radio button marked **Specify an IP Address** then enter the correct data for **IP Address** \*

and, if necessary, the **Subnet Mask** (this value must match the Subnet Mask on QuBit)

and **Default Gateway** \*

*\* If you do not know the values to enter for IP Address or Default Gateway, see your network admin.*

---

## Appendix C: Customizing QuBit through Environment Variables

*Environment variables* is the catch-all phrase used to describe a small area of memory that is used as a kind of scratchpad to remember certain kinds of frequently-used information. These pieces of information are *environmental* because they're about and affect the operating system environment as whole, and *variable* because they can change from system to system, from reboot to reboot, or at will—altered by a script, a program or by the human operator.

They include a variety of current QuBit settings such as the recording SNR, the image format, QuBit's IP address, mirror drive settings, and more. Knowing these settings you can customize QuBit's startup (for example, so that it is always ready for 57dB recording at 1920 x 1080 by 30i), as well as change its current operational state. You can use environment variables to put you in greater control of QuBit.

QuBit's environment variables are set during startup. You can change any or all of them from a telnet or serial shell. Some changes take effect immediately; others require a reboot to take effect. In the list that follows on the next page, values with an asterisk (\*) can be permanently saved to non-volatile memory so they are automatically "in place" on each startup. Others can be added to the custom.bat startup file so that they too are automatically set upon bootup.

Example: to change the network IP Address for your QuBit to 200.0.1.128, open a serial or telnet shell and type:

```
set network,ipaddress 200.0.1.128 save (then press Enter).
```

Now type:

```
reset (then press Enter. QuBit will restart with the new IP address in place.)
```

Although you can change IP settings from the front panel, not all environment settings can be modified there. Some may only be changed via shell commands, or by editing the QuBit's user-startup text file, custom.bat (see the appendix "Customizing QuBit Startup").

**Note:** *Changes to some environment variables (such as IP address) will not take effect until QuBit is restarted, while others take effect immediately.*

### Setting or Changing an Environment Variable

When you wish to change a variable, you tell QuBit to *set* the value for a specific variable to a new value. The general format for changing a variable is: `set groupname, <itemname> <value>`

For ease of use, QuBit's environment variables have been categorized into groups. This makes it easier to review the variables for a particular group without having to display the entire list. Once you become familiar with QuBit's environment variables, or at least the ones you find yourself changing most often, you can forego the group name and simply set the value directly, as in `set itemname <value>`

For example, to set the recording SNR you could type either `set video,lumsnr 54` or `set lumsnr 54` since QuBit will accept both forms of the command.

Variable settings may be incorporated into your own batch files, such as the custom.bat startup file, to customize QuBit to your taste.

**To see all of QuBit's variables in a single list** Type `show all` and press Enter. Make sure your serial shell or telnet shell preferences have a console buffer so that you can scroll back through the list, since it will fill several pages. The entire list is shown on the next page.

**To display all group names** Type `show groups` and press Enter. QuBit will respond with a list of the groups available, similar to this:

```
AUDIO
DEBUG
DMA
GENERAL
IO
MAGIC
MEMORY
NETWORK
PROTOCOL
SCSI
SERIAL
TIMECODE
VIDEO
```

**To display the settings for a particular group** Type `show` followed by the name of a particular group, and press Enter. EXAMPLE: To show all of the variables in the network group, type `show network` and press Enter. QuBit will respond with a list similar to this:

```
NETWORK
  IPaddress = 192.168.51.48
  SubnetMask = 255.255.248.0
  DefaultGateway = 192.168.48.1
  EthernetAddress = 00.60.FF.00.11.93
```

**To see the valid values for a specific parameter** Use a question mark. EXAMPLE: if you are not sure of the possible values for a clip's "hold value," type `set video,hold ?` QuBit will respond with a reply like this:

```
Legal values for video,hold are one of:
  Off
  Field
  Frame
```

**To set environment variables from the custom.bat startup script** Simply add the command in the same form that you would use from a shell. EXAMPLE: if you prefer to have QuBit startup in a ready-to-record state, in 1920x1080x30i HD, at 54 db SNR, with an open shell on serial port B (a useful trick, since it allows you both serial shell and access and a usable GUI display at the time), add these commands and variable settings to the text file `custom.bat` (in the lines marked for user customization):

```
hd2_1920x1080_30i      ; sets QuBit to 1080i HD image size and 30i display rate
vmode r               ; sets up record mode
set lumsnr 54         ; sets SNR to 54 dB
set anaoutcolorspace rgb ; set component output to RGB colorspace
shell serb            ; opens a shell for a serial connection on Serial Port B
```

## Group Names

Sample output of the show all command.

```

AUDIO
  Enable = TRUE
  Channels = 6
  Bits = 16
  SampleRate = 48
  AnalogPair = TRUE
  ClkSource = Internal
  Volume = 100
  XFadeClips = False

MEMORY
  CPU = 67108864
  HDMA = 67108864

NETWORK
  IPAddress = 192.168.51.74
  SubnetMask = 255.255.248.0
  DefaultGateway = 192.168.48.1
  EthernetAddress = 00.60.FF.00.10.74

VIDEO
  Tap = VSP
  TapWidth = 2
  MetaMode = FALSE
  Hold = Off
  IRQ = TRUE
  Mode = Play
  Rate = 24
  PixelClockOut = 74250000
  PixelClockIn = 74250000
  FrameTotalLines = 1125
  FrameSyncLines = 7,0
  FrameIntervallines = 34,0
  FrameActiveLines = 1080,0
  FrameBlankLines = 3,2
  FrameLines = 1114
  FramePixels = 1920
  FrameBufPixels = 1920
  StoreVertInterval = FALSE
  WindowLines = 1026
  WindowPixels = 1280
  Bits = 12
  Components = Y,Cb,Cr,none
  Bandwidths = 4,2,2,0
  Encoder = 1
  Transform = ON09
  LacePredict = TRUE
  Scanning = Progressive
  SNR = 45
  LumSNR = 45
  LossMapVer = 2
  CompVer = 2
  InputSrc = Parallel
  AnaOutSync = TriLevel
  AnaOutColorspace = YUV
  OutputClkSrc = Internal
  FrameLock = FALSE
  FrameLockDelay = 0
  AnaOut_Y_gain = 864
  AnaOut_CR_gain = 928
  AnaOut_CB_gain = 928
  AnaOut_Y_offset = 512
  AnaOut_CR_offset = 512
  AnaOut_CB_offset = 512

DEBUG
  Baud = 9600

DMA

GENERAL
  TimeZone = -600
  DaylightSavings = USA
  SerialNumber = 306
  BootName = qubit.bin
  DefaultBoard = A
  Mode = HD2_1280x1026_24p
  ModeShort = HDw_24p
  UserID = (none)

SCSI
  Mirror = False
  Delete = Fast

SERIAL
  SERa_Baud = 38400
  SERa_Flow = None
  SERb_Baud = 38400
  SERb_Flow = None

TIMECODE
  Record = External
  Output = Rec
  InDelayAdj = 0
  OutDelayAdj = 0
  LTCoutGain = 8
  IntDelayAdj = 0
  DropFrame = TRUE

GROUPS
  Audio
  Debug
  DMA
  General
  IO
  Magic
  Memory
  Network
  Protocol
  SCSI
  Serial
  Timecode
  Video

IO

MAGIC
  Enable = FALSE
  KillFile = TRUE
  NameFrom = Dir
  Dir =
  DestPath = BOOT:
    
```



**General Environment Variables**

private	boolean	Set TRUE to enable access to private (development only information)
AutoSetup	boolean	(not implemented)
* TimeZone	numeric	Hours East/West of prime meridian
* DaylightSavings	mnemonic	Disables for enables daylight savings rules ("NONE" or "USA")
* SerialNumber	numeric	Serial number of Qubit. Cannot be changed without the proper key
* ShellBaud	numeric	Selects the baud rate for the serial shell terminal (pick from 110, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600 or 115200)
* BootName	string	Specifies which build to run when booted (requires reboot)
DefaultBoard	mnemonic	Sets the default board, if none is specified ("A" "B" "C" or "D")
UserID	mnemonic	<p>A user-defined string of up to 8 characters (alpha, numeric, punctuation, anything) that is permanently stored in QuBit's non-volatile memory.</p> <p>This lets you distinguish one QuBit from another, using a labelling system of the your preference. There is a feature in scripts which lets you play different content depending on the setting of this variable.</p> <p>Originally intended to solve a "screen ID" problem so one master script and set of content would allow different watermarks and trailers play on different systems.</p>

### Memory Environment Variables

CPU	numeric	Read-only value which reflects the total amount of CPU memory
HDMA	numeric	Read-only value which reflects the total amount of HDMA memory

### Video Environment Variables

Tap	mnemonic	Selects the video board and tap to use during record/play. One of "DIO","VIO","VSPa","VSPb","VSPhdma". Currently usually set as part of a batch file that handles setup for a particular mode.
AutoColor	boolean	(DIO only) automatically adjust hardware to match colorspace of clip
MetaMode	boolean	Special "Metavision" mode
PathManage	boolean	Do extra control of video hardware to manage data flow (required) wFlipFlds boolean(obsolete)
Hold	mnemonic	"OFF", "FIELD", or "FRAME" (for DIO only). Determines output when a clip is stops playing. OFF is a gray screen. FRAME holds last image of clip. FIELD not supported.
MaxCatchup	numeric	Max limit of catchup rate for video that falls behind due to SCSI errors (0 to disable)
FieldSync	boolean	Coerce video to correct fields, if out of sync (DIO only)
IRQ	boolean	Enables/disables the video interrupt
DryUp	numeric	(Do not use, for testing only)

**Video Environment Variables (continued from previous page)**

SlowDown	numeric	(Do not use, for testing only)
DMAmap	numeric array	DMA channel to use for each channel (up to 4). Set each to -1 to disable
ArrayOrder	numeric array	Logical to physical drive mapping for video array
FieldOrder	numeric	Field XOR fudge to compensate for DIO problems
Mode	mnemonic	"NONE", "PLAY", OR "REC" (mirror of the Vmode command status)
Lines	numeric	Number of lines per frame
Pixels	numeric	Number of pixels per line
Bits	numeric	Number of bits per pixel (should be multiple of 8)
Rate	numeric	Sets the video frame rate, in frames per second. Valid values are 23.98, 24, 25, 29.97 and 30. Note that 50, 59.94, and 60, which are field rates, are considered valid by the system, but not currently supported. Although this parameter can be changed at any time, the change will not take effect until the next time the record or play mode is entered. The easiest way to accomplish this is to do a <i>vmode n</i> command , followed by <i>set video,rate &lt;value&gt;</i> , then <i>vmode r</i> or <i>p</i> .
FrameSpan	numeric	Number of frames required to transfer a single frame (0 to disable mode)
Quantdata		
Bandwidths		
Encoder		

### Video Environment Variables (continued from previous page)

Transform		
Lacepredict		
LumSNR or SNR		Used to set the quality level of the recording, based on the signal-to-noise ratio (SNR). Not used in play mode. Values from 36-69 are currently supported, but not by all modes.
AudioDelay		Works on record and playback. Limited to 30 frames delay.
InputSrc		<p>Selects the incoming signal source (not used in play mode).</p> <p>The default is <b>Parallel</b> for the parallel digital input. <b>SerialSD</b> selects the Standard Definition serial digital input. <b>SerialHD</b> selects the High Definition serial digital input.</p> <p>Although this parameter can be changed at any time, the change will not take effect until the next time QuBit enters record or play mode. To accomplish this is to set the <code>inputsrc</code>, <code>set video,inputsrc &lt;value&gt;</code>, then set the image mode, and finally set the QuBit to record mode <code>vmode r</code>.</p>
AnaOutSync		Selects the type of sync signal on the analog (RGB/YUV) output. Choices are <b>BiLevel</b> , <b>TriLevel</b> , or <b>Ext_H</b> . Usually you must choose one of these to accommodate a particular display such as a projector that requires a specific form of sync.
AnaOutColorSpace		Selects the format for the color space of the imagery on the analog (RGB/YUV) output. Choices are <b>YUV</b> or <b>RGB</b> .
OutputClkSrc		Selects the proper timing signal to be used as a master clock for playback. Playback will be sync'd to the choice you make here. Choices include <b>Internal</b> clock, <b>Parallel</b> input, <b>SerialSD</b> input, or <b>SerialHD</b> input.
FrameLock		Affirmative choices include <b>T</b> , <b>True</b> , <b>Y</b> , <b>Yes</b> , or <b>1</b> . Negative choices may be <b>F</b> , <b>False</b> , <b>N</b> , <b>No</b> , or <b>0</b> .
FrameLockDelay		Legal values are 32-bit positive decimal values.

**Video Environment Variables (continued from previous page)**

AnaOut_Y_Gain		Legal values are 16-bit positive decimal values
---------------	--	---

AnaOut_CR_gain		Legal values are 16-bit positive decimal values
----------------	--	---

AnaOut_CB_Gain		Legal values are 16-bit positive decimal values
----------------	--	---

AnaOut_Y_Offset		Legal values are 16-bit positive decimal values
-----------------	--	---

AnaOut_CR_Offset		Legal values are 16-bit positive decimal values
------------------	--	---

AnaOut_CB_Offset		Legal values are 16-bit positive decimal values
------------------	--	---

### Audio Environment Variables

AnalogPair	boolean	Selects between the analog and AES/EBU digital inputs for channels 1 and 2 in record mode; not used in play mode. During play back, the data is output on both the analog and digital outputs, regardless of its original source. Set this parameter to "true" for analog, "false" for digital.
Bits	numeric	Determines the resolution of the audio data to be stored, in bits per sample. 16, 18, 20 or 24 bits per sample may be selected. If the incoming digital audio data is a different resolution, it will be rounded to match the selected resolution.
Channels	numeric	The resolution of the audio data to be stored, in bits per sample. Not used in play mode. 16, 18, 20 or 24 bits per sample may be selected. If the incoming digital audio data is a different resolution, the data will be rounded to match the selected resolution.
ClkSource		The sample clock source for incoming audio data in record mode. "Internal" selects the internally-generated clock. If only analog data is being recorded, this parameter must be set to internal. Digital data may optionally be recorded using the internal clocks. "Digital12" and "Digital34" selects the clocks recovered from the AES/EBU digital inputs 1&2 or 3&4, respectively. Only one clock source may be selected. All other inputs are clocked using the selected source.
DMAchannel	numeric	(do not adjust)
Enable	boolean	Enable or disable audio processing
SampleRate	numeric	The incoming sample rate, in kHz. Currently supports 48 (the system considers 22.05, 32, 44.1 and 48 valid; the hardware only supports 48 at this time). When recording analog data or using the internal clocks to record digital data, the sample clock is set based on this parameter. When recording digital data using the recovered external clock, the data is tagged with this parameter but actually sampled based on the recovered clock. In this case, if the SampleRate parameter is not set correctly on recording, it will cause audio playback problems because the system uses the information stored with the clip to set the play back sample rate.
Volume	numeric	Sets relative volume level of the audio output, from 0 to 100. Affects all channels and may be changed while playing.
Xfadeclips	boolean	Performs a small "crossfade" between audio clips when played back to back.

**Time Code Environment Variables**

DropFrame	boolean	Currently undocumented. Awaiting programmer's notes.
IntDelayAdj		Calibrates the timing of timecode reported to controllers thru 422 control ports. Does not affect timing of LTC out the back of the Qubit. Should never be adjusted.
LTCOutGain	mnemonic	Sets the gain of the LTC output. Valid ranges are 0-15, and the default is 8. Because there seem to be several different standards in use, this parameter allows the user to adjust the output gain, if needed, in their particular set-up. For most users, the default value will work correctly.
Output	mnemonic	<p>Selects timecode to output during clip playback</p> <ul style="list-style-type: none"> <li>"NONE" outputs nothing</li> <li>"CLOCK" outputs QuBit video clock</li> <li>"REC" outputs timecode recorded in clip</li> <li>"CLIP" outputs clip elapsed time</li> </ul> <p>"None" and "clock" are the same as described above. "Rec" outputs the timecode that was recorded in the clip. "Clip" outputs a timecode generated by starting at 00:00:00:00 at the beginning of a clip.</p>
Record	mnemonic	<p>Selects timecode to be recorded with a clip</p> <ul style="list-style-type: none"> <li>"NONE" records none</li> <li>"CLOCK" records timecode from the QuBit video clock</li> <li>"EXTERNAL" records incoming timecode</li> </ul> <p>Determines which timecode source is recorded; not used in play mode. No timecode information is recorded when "none" is selected. The timecode generated from the internal, free-running clock is selected by "clock". The timecode that is embedded in the clip starts at the current value of that clock when the recording starts. "External" selects the external timecode input as the source.</p>
SyncClock	boolean	Synchronize Qubit video clock to external timecode (LTC input)

### SCSI Environment Variables

Delete	mnemonic	<p>Governs how the DEL command is implemented. Default mode for this setting is FAST (files are only marked “free,” data remains recoverable until overwritten by new data). Set to SECURE so that files are overwritten with 00's and no longer recoverable. To permanently configure QuBit for secure deletes, set the SCSI,DELETE variable appropriately in the Qubit's CUSTOM.BAT file.</p> <p>Note that this is not a DoD 5220.22-M level erase, but you cannot recover it with the RECOVER command. The data is gone.</p>
Mirror	boolean	Set to “true” to make a mirror drive set active for playback. Set to “false” if there is no mirror drive set or to turn mirroring off (while performing diagnostics, for example).
TestMode	numeric	(Do not use, for test only)
SyncPeriod	numeric	Usually 100 (ns) for max rate of 10 MW/s
* Termination	mnemonic	Sets termination on the CPU board for each channel. Can be “OFF” or “ON” to set them all the same, or “——” substituting a ‘T’ for each ‘-’ to turn termination on for that channel (channel 0 first)

### DeBug Environment Variables

Serial	boolean	(Do not use)
ForceRetryType	numeric	(Do not use)
ForceRetryWait	numeric	(Do not use)
VidThreshLo	numeric	Size of video DMA, in bytes, at which to flag an error (low side)
VidThreshHi	numeric	Size of video DMA, in bytes, at which to flag an error (high side)
VidLogDepth	numeric	Depth of video log (0 to disable)

**Network Environment Variables**

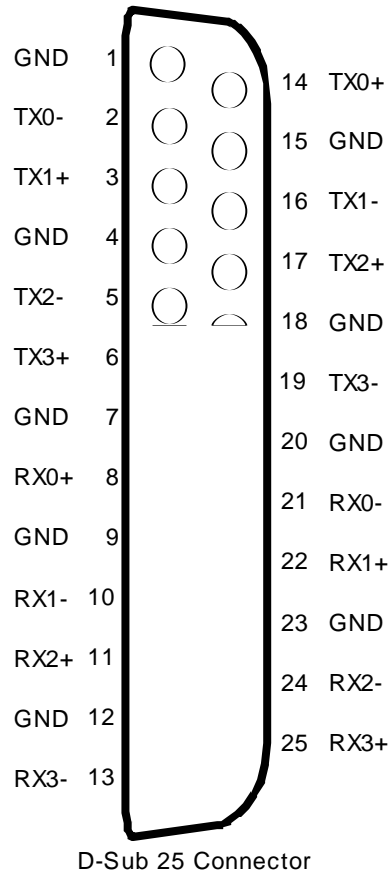
* IPAddress	numeric array	Ethernet IP address of machine, in "dot" format (e.g. 200.0.1.55)
* SubnetMask	numeric array	Subnet mask, in "dot" format
* DefaultGateway	numeric array	IP address (in "dot" format) of default router
* EthernetAddress	numeric array	Physical ethernet address of machine (6 hex values in "dot" format). Should be 00.60.FF.xx.xx.xx for QuVIS devices (allocated to us by IEEE)
MaxFTPconns	numeric	Maximum number of users on FTP server simultaneously (only supports as of this writing)

**Magic DirectoryEnvironment Variables**

Enable	boolean	Enable the "magic" auto-processing feature or not
NameFrom	mnemonic	"DIR" to select name of clip based on path to magic file "FILE" to select name of clip based on name of magic file (sans frame numbers)
Dir	string	Path/name of "magic" directory
DestPath	string	Path in which to create clips via "magic" auto-processing



### Appendix D: Pinout Diagram for Digital Audio Connector

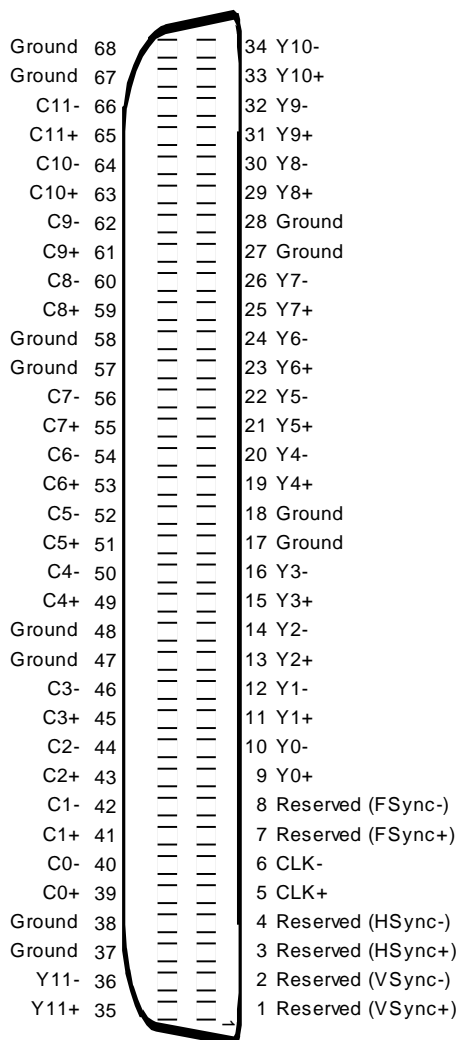


**Channel Assignments:**

<u>Channel #</u>	<u>Receiver/Transmitter</u>
1 and 2	RX0/TX0
3 and 4	RX1/TX1
5 and 6	RX2/TX2
7 and 8	RX3/TX3



### Appendix E: Pinout Diagram for both Parallel Connector Input and Output



Notes:

1. Reserved pins should be left unconnected.
2. Bit number 11 is the most significant bit. 10-bit systems should use Y11 - Y2 and C11 - C2.



## Appendix F: Updating Your QuBit Operating System Software

As QuVIS releases updated system software for QuBit, you may download these new releases from the support.quvis.com web site and install them for improved features and performance. To update your QuBit OS, you will need an internet connection (to download the files), and an Ethernet connection to QuBit (to transfer the files to QuBit, and to install them).

New releases are currently located in the support site's FTP File Area, and currently consist of two files: `system.rls` and `getrls.mod`. `System.rls` contains the fully updated system software, while `getrls.mod` is the software module used to install it.

### Updating the System Software

Step 1: Download the files from support.quvis.com. Make sure you know what release you are downloading.

Step 2: By FTP, transfer the `system.rls` and `getrls.mod` files to the root directory of QuBit.

Step 3: Open a serial or telnet shell to QuBit and type `getrls system.rls` to install the update. The current system directory will be renamed `system_old`, and a new system directory will be created for the updated operating system. Custom files in `/sys/user` will remain in the active system directory, as will pre-existing security files.

Step 4: Test the new release as desired to ensure compatibility.

You may wish to create a new directory called "releases" in which to store the `system.rls` file, and other older system software updates, in case you wish to go back to an older release.

Also, if you prefer, you may wish to rename or move the `system_old` folder to another directory in case you wish to restore it.

---

#### ***Important!***

If you have modified or created any files in the system directory, be aware that you may need to retrieve them from the renamed `system_old` directory following a system software update.

---

### Restoring the Previous System Software

Should you need to go back to the prior release of system software, simply rename the current system directory to something else, such as `system.xxx`. Then rename the `system_old` directory to `system`. Reboot QuBit, and the previous release will be back in effect.

### Tips

In the QuBit shell:

`dir -v` will tell you the version of `system.rls` installed;

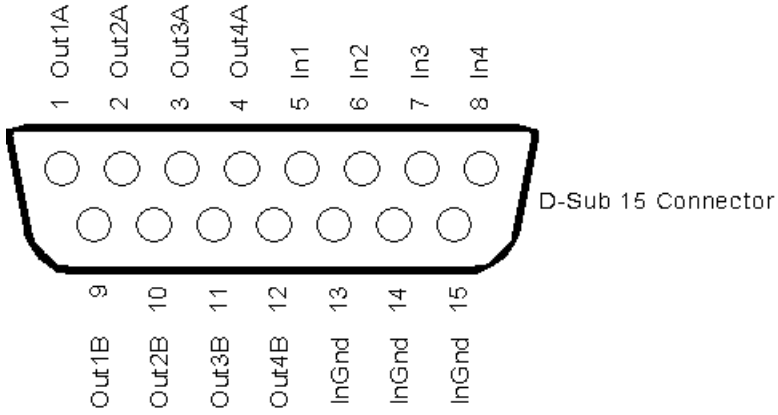
`ver rls` will tell you the version of all `.rls` files installed on a QuBit.

A DOS utility called `showver.exe` can query the version number of a file on a PC. You'll find it on many public domain and shareware file sites on the Internet.

---



### Appendix G: Pinout Diagram for GPI



Note:  
Input Grounds are isolated from the rest of the system



## Appendix H: The QuBit Shell (QShell)

The QShell prompt is always the current directory, followed by the prompt symbol: >. All commands in this shell are QuBit-specific. For ease of learning, they bear some resemblance to “typical” shell commands. As with DOS commands, some are useful in-the-field; others are only useful for testing/debugging.

### Communicating with QuBit

The QShell is normally provided via RS-232 serial to a “dumb” terminal or a computer running a communication or modem program. The QuBit serial port to use for this is labeled RS-232 (A). The settings on the computer end should be 8 bits, no parity, 1 stop bit. The baud rate should be set to match the value set in the boot menu (usually 38400).

The QuBit also contains a Telnet server, which is tied into the QShell. Notice that to delete characters properly with Intergraph's Telnet client (under Windows 95), you must change setup/keyboard/key behavior: Backspace to use backspace, not delete.

### Commands in the QShell

There are three types of commands that can be invoked from the QShell: internal commands, loadable code modules (.mod files), and batch (.bat) files. The search order used to interpret and execute a given command is as follows:

- ❖ Search internal commands (see below)
- ❖ Search for name.mod in the current directory
- ❖ Search for name.mod in the /sys/mods/ directory of each drive on the search path
- ❖ Search for name.bat in the current directory
- ❖ Search for name.bat in the /sys/bat directory of each drive on the search path

The search path is usually:

- ❖ The “boot” drive
- ❖ Any other drives added in the search path added by the path command

### Entering Default Data Types

Most internal and module commands obey the same rules of data types assumed on the command arguments. In most cases, you may specify numeric arguments in whatever base is most natural for you. Decimal is usually assumed, unless you precede the number with \$ or 0x which makes them Hexadecimal. (The parser is smart enough to assume hex even if you forget these, as long as the number contains some of the hex alpha characters – it's not good practice to be this vague, though.)

Exceptions to this rule are the internal mm and dm commands (see below), which assume hexadecimal (for convenience).

### Using the Command History

The QShell maintains a 20-line history buffer. By using the up/down arrow keys, you can browse through this buffer looking for a recent command line that you would like to invoke again. Press Enter to invoke the displayed line again, or you may optionally edit the line first. Every command line, when executed, is added to the bottom of the history buffer, and the oldest line is lost.

Note that for this feature to work, your terminal must send ANSI sequences in response the up/down arrow keys. Try using ANSI or VT-100 terminal emulation. If running under Windows, ensure that the terminal preferences option “Use Function/Error, and Ctrl Keys for Windows” is not checked.

### Software Modules

Software modules are actually small programs that can be run from the command line (or from a batch file), much like executable files in other operating systems such as MS-DOS. Modules are binary files with a .mod extensions. They should all be placed into the sys/mods subdirectory. To invoke a module, you simply type its name (without the .mod extension).

### Batch Files (aka Batch Text Files)

Batch files are text files with a .bat extension. Generally they should be placed into the sys/bat subdirectory (if they affect system-wide operations) or the clips directory (if they are playlists). To invoke a batch file from the QShell, you simply type its name without the .bat extension. To invoke a batch file from the LCD GUI, select and play the bat file as if it were a clip.

Batch files may contain any commands that you can use at the QShell, including internal commands, loadable modules, and other batch files.

Arguments may be passed to batch files. They 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. %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 have been passed to the file, otherwise an error is reported and none of the file is executed.

Batch files may be nested (one may invoke another). The depth of nesting is only limited by the maximum number of open files (which is currently a fixed number).

Comments may be placed in batch files by using a semicolon (;) in the first column. The entire line is then considered a comment. Delays may be added using the internal Wait command (see below). Currently, no looping or conditional primitives have been implemented.

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 (this may be a user option in the future).

### If All Else Fails...

Should you forget the commands or the command line patterns, you have two levels of built-in help available.

First, type `help` to see the full list of shell commands available to you.

Second, most commands have built-in help templates that further show how they are used, or the arguments and options you can use with them. Type the name of any command all alone on the command line. QuBit will then display information about how the command is used.

*continued on next page*

## Internal Commands

QuBit's operating system includes a set of commands and internal modules (which you run like commands) for basic navigation, file maintenance, and operations.

### Assign

---

**Example:** *assign* [*< name >*]

*assign* *< name >* *<path>* **ADD**

*assign* *< name >* *<remove>*

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.

**path** Any directory path. For example 287D:clips\recent\logo\tests could be assigned to simply "tests" by typing `assign tests 287D:clips\recent\logo\tests ADD`

### Atten

---

**Usage:** *Atten* *<chan>* [*<value>*]

*or* *Atten* **TRIM**

Adjust audio attenuator settings. Allows inspection and/or modification of the audio attenuator settings. The channel values may be 1 through 4 for the four analog inputs, or 5 for the headphone jack. If a value is in fact specified, the attenuator specified will be adjusted to this level. If it is omitted, the current value of the specified attenuator is displayed. The TRIM option displays a table of these values which may be navigated and altered (only for ANSI-compatible terminals).

### Auth

---

**Example:** *auth* *<key>*

Protects clips unauthorized playback without a proper password. Allows user to enter a password (done before playclip is attempted). After four hours have passed the password expires and is no longer valid.

---

**BatchMake**

**Example:** *batchmake* [*< path >*] *< clipname >* [*KILL | DATES*]

Runs the batch file *magic.bat* for each file in the directory specified by the path (*boot:/magic/* is the default). Files are processed alpha-numerically. To stop the BatchMake process prematurely, press Ctrl-D at the console. It will terminate at the completion of the current invocation of *magic.bat*.

*Note:* The default *magic.bat* invokes the *makeclip* function to merge frames into a clip. In actuality, anything useful can be done en masse to a directory of sorted files using this mechanism.

path	the directory containing images to be processed
clipname	the name of the resulting clip
DATES	process files in the order of oldest to newest
KILL	deletes each file after the batch file processes it

**Browse**

**Usage:** *browse* *<clip>*

< Prev data type  
 > Next data type  
 4 Prev frame  
 6 Next frame  
 ! Set frame jump to 1  
 @ Set frame jump to 2  
 # Set frame jump to 3  
 \$ Set frame jump to 4  
 8 Move up 1 line in data  
 2 Move down 1 line in data  
 9 Move up 1 page in data  
 3 Move down 1 page in data  
 7 Move to top of data  
 1 Move to bot of data  
 5 Move to middle of data  
 D Enter data position  
 F Enter frame number  
 S Save component to file  
 L Load component from file  
 P Preview current frame to video  
 V Toggle continuous view mode  
 Space or Enter to exit.

**CD**

**Usage:** *cd* [*<path>*]

Sets the current working directory to the path specified. If no path is specified, simply displays the current directory.

path                specifies the current working directory

**CineGUI**

**Usage:** *cinogui*

CineGUI is used to run cinema material. CineGUI is documented in Appendix P. (Not all QuBits are capable of running CineGUI. A DVD drive is required.)

**CMPclips**

**Usage:** *cmpclips* *<origclip>* *<testclip>* [*-options*]

Compares data from two clips to determine if they are identical. Options include:

- s        start frame to process
- f        number of frames to process
- q        quit on first error
- d        details

**Clock**

**Usage:** *clock show* [*UTC*]

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

Displays the current time as set within QuBit or sets the built-in real-time-clock to the date and time you wish.

**Comment**

**Usage:** *comment* *<name>* *<comment>*

Adds a comment to a file.

**Config**

**Usage:** *config* *<file>*

Undocumented.

**Copy**

**Usage:** *copy* <srcfile> <destfile>

This command copies a file from one location to another. The original file remains in its original location.

**Notes:** *The current path is applied, so if you wish to copy files from other directories, make sure to include the full path to them. Also, you can specify a new name for the copied file (if you simply wish to move it to a new directory location and rename it in one step, use the ren command).*

**CopyData**

**Usage:** *copydata* <srcdrive> <destdrive> [options]

Raw drive copier. Options include:

START <lba>	First block to copy from source drive(s)
END <lba>	End of area to copy from source drive(s)
DEST <lba>	Directs to different blocks on dest drive(s)
ARRAY	Does identical operation on entire drive array simultaneously
VERIFY	Compare data between source and dest drive(s)
NOCOPY	Skip copy operation
LOG	Write results to log file
DETAILS	List all verify errors (very verbose)

**DACtest**

**Usage:** *dactest TRIM*

*dactest* <group> <chan/ALL> <value> [sub]

Adjusts DAC settings. Exit by pressing ESC twice. Must be done from a serial shell connection *only*. A telnet type of shell does not allow for navigation within the table that *dactest trim* displays.

**Del**

**Usage:** *del* <path/filename> [-q] [-noquery]

Deletes the specified file, directory, or clip.

-q	Ask about each item to be deleted.
-noquery	Do not ask about any items that are to be deleted.

**DelTree**

**Usage:** *deltree* <directory>

Deletes a directory with all its children (subdirectories and files therein).

---

**Dir**

**Usage:** *dir* [<path>] [-ALL] [-ALT] [-REV]

Lists the contents of the specified directory, along with details about each file, directory, or clip. If no path is specified, uses the current directory. The optional ALL keyword causes all sub-directories below the current directory to be displayed as well. The ALT option displays various internal details about each file, clip, or directory. REV option displays revision information, if possible, about each file displayed.

*Note: the keywords ALL, ALT, and REV now should be preceded by a hyphen to distinguish them from subdirectories that may be named "all," "alt," or "rev."*

---

**DispLib**

**Usage:** *displib* <operation> <arguments...>

Various tests of the display.lib.

---

**Exit**

**Usage:** *exit*

Quit and close the current shell or telnet shell connection.

---

**GetFault**

**Usage:** *getfault*

Used in conjunction with the fault GPI output. When a major fault is detected (whether recoverable or not), the fault GPI output goes active. The GetFault command can then be used to retrieve the fault code, description, and any additional info. This action also clears the fault code and clears the GPI output, hence acknowledging the fault. It is permissible to use this command without using the GPI outputs. If no fault has occurred, it will simply return fault code 0 (Okay).

---

**GetTC**

**Usage:** *gettc*

Displays the current time code settings for QuBit. The format is shown below. "None" indicates no value set or input present; otherwise a time code value will be displayed.

LTC in: (none)

LTC out: (none)

Internal: HH:MM:SS:FF

---

**Extract**

**Usage:** *extract* <clip> <startframe>[:<endframe>] <output file> [TGA|YUV|QVS]

This tool extracts a specified still frame out of a video clip. Clip is the name of the clip, plus a path if necessary. Startframe is the number of the frame to extract, from 1 to N, where N is the length of the clip (in frames). Output file is the name (plus path, perhaps) of the still frame to generate. The last argument specifies the format of this frame. Targa is the only currently supported option.

**Flash**

**Flash** <PROG|VER> <file>

**Flash** <ADD|CHK> <file> <flash file name>

**Flash** <DEL> <flash file name>

**Flash** <REN> <flash file> <new flash file>

**Flash** DIR

Reprogram the FLASH EPROMs from a hex file.

PROG is used to program the boot ROM

VER verifies the boot ROM with a file

ADD|DEL are used to add or delete files from the storage ROM

REN renames a file in the storage ROM

CHK verifies a file in the storage ROM

DIR shows the files that are in the storage ROM

**Getrls**

**Usage:** *getrls* [-h|-?] <rls file>

<bin file> is the input file to that contains the build.

Where: -h|-? prints this message and exits

<rls file> is the input file to that contains the build.

**Note:** *More than one build file can be entered. Builds will be extracted in the order listed.*

**GUI**

**Usage:** *gui*

This mod runs the standard QuBit GUI that you see on the LCD front panel. If you have a serial terminal attached to QuBit during startup, you need to press ctrl-D to exit the GUI and activate the shell. You can re-run the GUI by typing this command from the shell command line.

---

**HDDiag**

**Usage:** *hddiag* <drive> <lba start> [<lba end>] [-v] [-r] [-w] [-s] [-i]

**Important Notes:** You must issue the command `vmode n` to have the available HDMA memory to run the above tests. Also, at least one of `-v`, `-r`, or `-w` must be used. The `-r` and `-w` options may not be supported on all drives, the program should correctly report this if it occurs.

Using `hddiag` you can test and verify any section of a QuBit hard drive using built-in SCSI commands. This should help in identifying problems with a drive. `Hddiag` changes some drive settings temporarily, and sets them back when finished. If for some reason the program can't set the values back, a reset should restore them.

The **start lba** and **end lba** parameters allow you to verify a portion of the drive. The start lba must be specified, however if the end lba is not specified the maximum lba is used. For example, if you enter `lba start 0` the entire drive will be tested.

Options:

- v Verify drive using drive's ECC data, this is the fastest test.
- r Verify drive using a byte compare of the data that is read from the drive.
- w Verify drive by writing and verifying the data that was read from the drive. [DISABLED] The write verify option [-w] has been disabled as some IBM drives have problems with it. It should hopefully be returned soon. Until then the destructive write verify option is both the most robust test (and most dangerous to the drive).
- i Ignore non-SCSI Check Condition errors. Test will attempt to complete no matter what errors occur (except SCSI Check Condition).
- s Verbose statistic mode. If an error is encountered, it is printed out along with the sense request data bits, with out this options, only a summary is printed at the end of the test.

---

**Help or ?**

**Usage:** *help*

?

Displays a quick summary of the QShell's internal commands.

---

**Hmemtest**

**Usage:** *hmemtest*

Performs a test of QuBit HDMA.

---

---

**Jog**

**Usage:** *playclip <clipname> shuttle ENTER, then jog <framenumber>*

**jog stop**

Jog is available when you play a clip with the shuttle option. Jog allows you jump to any frame in the current clip, and from there to any other frame. To use jog, play a clip with the shuttle option:

```
playclip kansas_postcard shuttle
```

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

```
jog 6000
```

Once in this virtual jog mode, you can continue to enter new frame numbers to jog to without having to re-enter the entire line. Use `jog stop` to exit jog mode from the command line. Afterward, you may use the shuttle options and scan through the clip (see Shuttle for details).

---

**Log**

**Usage:** *LOG <classes> [ADD] [OLD] [CHECK]*

System errors and events are sent to a new system in the OS called the "log". Each item will have a class (drive error, video failure, overtemp, etc), a severity (comment, warning, error, etc), a time-stamp, an English description, and possibly other info. Currently, only the "DRIVES" class is supported. If the log ever becomes full, oldest messages are removed to make room for new ones. Currently, the most recent 100 messages (limited to 40 characters) are kept in the log, and the log is not cleared by a reboot.

Reading the log is currently implemented using the following command:

```
LOG <classes> [OLD] [CHECK]
```

The typical log message response shows date, time, class, severity, error code, a message, and extra drive info (such as the drive that failed, shown in the case below).

```
04-26-2000 05:11:29 Drives Error 8302 Play read error 30
```

To display only new messages (that have not been read previously), type the following:

```
log drives
```

To display all messages in the log (even those previously read), use the OLD option:

```
log drives old
```

To report the number of new entries in the log, but not view these entries, use the CHECK option (combine CHECK and OLD to report the total number of messages [old + new]):

```
log drives check  
log drives check old
```

A comment can be manually added to the log using the following variation:

```
LOG ADD "This is a comment"
```

---

---

**Login**

**Usage:** *login* <user> [*<password>*]

The standard QuBit, as shipped, has its security protection disabled. This allows any user to modify its operation freely, with complete access. However, if privileges have been set up on QuBit, you will need to provide a login name and password when you wish to work with QuBit.

The standard user, the technician, and the supervisor each have different degrees of control over QuBit. See your QuBit supervisor/administrator for your login name and password. See Appendix S for details on security setup.

---

**Logout**

**Usage:** *logout*

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

---

**MakeClip**

**Usage:** *makeclip* <filename> <clipname> [*<X offset>*] [*<options>...*]

Convert a Targa or SGI RGB file into YCrCb format and append field(s) to a clip. The only option for *makeclip* is TIME.

---

**Mirror**

**Usage:** *mirror* <clip> [*check*]

If you have opted to use a “mirror drive set” on QuBit, use the mirror command to clone individual clips from one drive set to the other. Clips are copied from the primary drive set to the mirror drive set, retaining their exact drive location in each set. Therefore, you cannot “reorganize” clips by copying them one at a time from the primary to the mirror. The mirror drive set will end up with the same file organization as the original set.

Use the [*check*] option to verify that the cloned data is an exact duplicate of the original.

To mirror the main drives to the mirror set use the *mirrordrives* batch file described in Appendix M.

---

**MkDir**

**Usage:** *mkdir* <name>

Creates a subdirectory within the current directory.

---

---

**ModClip**

**Usage:** *modclip* <inclip> <outclip> [-options]

Modifies clip in various user-selectable ways. You'll find it fully documented, with examples, in the Operations section of the main User Guide under "Adding Audio Channel Space to a Clip".

Options:

-a,	-audchans n	Number of audio channels
-b,	-bits n	Bits per audio sample
-q,	-samprate n	Audio samples per second
-s,	-start n	Start frame to process
-f,	-frame n	Number of frames to process
-r,	-rate n	New frame rate
-d,	-drop n	New drop-frame

---

**ModePage**

**Usage:** *ModePage* <drive> <page> *DUMP*

*ModePage* <drive> <page> <byte> <data>

Display/modify SCSI mode page data.

---

**Mount**

**Usage:** *mount* <drivenum>

Mounts the drive specified. Notice that this is a drive number, as in 0, rather than a drive name, as in "4.0." The drive must already have been formatted for QuBit before mounting with this command.

---

**Net**

**Usage:** *net mount ftp*

*net mount telnet*

*net stats*

The net command allows you to mount, or start, the QuBit ftp and/or telnet servers. Also, you can get statistics about the network. Here's an example of the output from the net stats command.

Receive stats:

Total Packets: 1203  
 Broadcasts: 591  
 Overruns: 0

Transmit stats:

Total Packets: 1053  
 Broadcasts: 0

Link status: 10

Duplex: half

Address	Port	User
-----	----	----
192.168.49.37,2838	Telnet	(none)
1 total connections		

**NewMC**

**Usage:** *newmc <image> <clip>*

Module which knows how to append QVS format frames onto a Qubit clip. Used internally by the OS.

**Path**

**Usage:** *path [<path> [ADD]]*

This command controls the building of the QuBit's search path list. If the ADD option is not specified, the path listed becomes the only path in the search path list. With ADD, the specified path is appended to the list.

## PlayClip

**Usage:** *playclip* <name> [<inpoint> <frames>] [<options>]

Starts playing the named clip and returns shell control.

Inpoint starts playback at the frame specified, otherwise starts at the clip's beginning.

Frames is the number of frames to play. If omitted, the entire clip is played. However, if an inpoint is specified, you **MUST** specify the number of frames also; use 0 to play from the inpoint to the end of the clip. Use the *stopclip* command or the STOP button on the QuBit front panel to stop prior to the end of the clip.

Options include:

loop	play the specified clip repeatedly until <b>CTRL-C</b> is entered at the console.		
queue	PlayClip command finishes when the clip is queued for playback (almost immediately). May not be used in conjunction with the ASYNC or SYNC options.		
async	Command finishes when playback of the clip begins. This is the default if none of QUEUEC, SYNC, or ASYNC are specified. May not be used in conjunction with the QUEUEC or SYNC options.		
sync	Command finishes when this clip finishes playback. May not be used in conjunction with the QUEUEC or ASYNC options.		
speed x/x	Specifies the playback speed and direction. Negative numbers cause reverse playback, from the outpoint specified to the inpoint. Positive numbers cause forward playback, from the inpoint to the outpoint. The number (x) can be a whole number or a fraction. Some examples will clarify:		

0	No motion (for use with SHUTTLE option only, does not output a still)		
1/3	Forward play at 1/3 x	-1/2	Reverse play at 1/2 x
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		

**Notes:** Speed specified is the initial speed and direction of playback. All numbers are positive (forward playback) unless a minus sign is present. Fractional speeds do not always have to be less than one. Minimum speed (in either direction) is 1/250 and maximum speed is 250.

Except in conjunction with the SHUTTLE option, 0 is not allowed. In SHUTTLE mode this can be changed during playback. Otherwise, this speed and direction cannot be changed until the clip ends (or is stopped using STOPCLIP).

(continued on next page)

shuttle	Plays the clip in “shuttle” mode. In this mode, the <i>SHUTTLE</i> command can be used to change the speed and/or direction of playback at any time. In this mode, the clip will stay in playback, even if one of the endpoints is reached. The video/audio hardware will be “in use” by this same clip until this special shuttle mode is exited, using the shell command <i>SHUTTLE STOP</i> . The speed specified using the <i>SHUTTLE</i> command are exactly the same format as those in the table above (positive or negative, integer or fractional). Specifying a speed of 0 will stop playback at the current frame. You must subsequently resume playback by setting to a non-0 speed, or end the mode using <i>SHUTTLE STOP</i> . Further <i>PlayClip</i> commands cannot begin until this shuttle mode is exited. Beware of the fact that a simple <i>PlayClip</i> command defaults to ASYNC, which waits for playback to begin. This will effectively tie up your shell and make it impossible to do a <i>SHUTTLE STOP</i> except from some other shell.
noaudio	Specifies that no audio should be played along with the video. Disables audio play back even if the clip was originally recorded with audio.
stats	Provides console display of statistics during playback, and a summary note. On the terminal, displays statistics about the clip while playing. The statistics that are of interest to the user include Err (the number of dropped frames or other errors), TC (the current timecode), and Dat (the total amount of data consumed to that point).
ltc sys <hh:mm:ss:ff>	Selects the linear time code at which to begin playback. When the incoming timecode matches the <hh:mm:ss:ff> specified, the clip begins. If this parameter or the next one is not included, playback begins immediately. Note that the timecode that is selected to be output (via the <i>set timecode, output</i> command) does NOT have to be the same as the timecode source that triggers the start of playback.
SYS <hh:mm:ss:ff>	Selects the internal, free-running timecode clock to trigger the start of playback. If this parameter or the previous one is not included, playback begins immediately.
hold	not documented

To stop the clip manually, use the `stopclip` command or the **STOP** button on the front panel.

## PlayList

**Usage:** *playlist* <file> [CACHE]

Parses the text playlist specified, and begins playing it. The format of a playlist is very simple. Each line must be either a comment, a clip to be played, or a control word.

```
; this is a comment.
; in the example below the first field lists the name of the clip.
; the second field specifies the length of playback, starting at the clip's beginning.
FlyOut          120
```

```
; in the example below the second field specifies the length of playback
; starting at the clip's beginning. Here the two numeric fields are the inpoint
; and outpoint (in frames), respectively.
FlyOut          30      150
```

```
; these are the possible control words
*LOOP*          10
*WAIT*          60
*SMPTE* 00:00:00:00
```

Notice that the control keywords must be surrounded by asterisks to distinguish them from clips named LOOP, WAIT, etc. The LOOP control word causes play to repeat back to the beginning of the list the number of times specified (0 goes forever). The WAIT control word causes a black pause of the specified number of fields to be inserted. The SMPTE control word turns on SMPTE timecode output, and starts counting at the timecode specified. When the playlist exits, the timecode stops.

Press **CTRL-C** from the console to stop immediately. Use **CTRL-D** will stop at the next completed clip.

The inpoints rather than actually playing the list. This is not useful in conjunction with a subsequent PlayList command, but with sequences of PlayClip's.

The current limit for a playlist is 200 entries (including clips and control lines).

## PostDub

**Usage:** *postdub* <clip> <audiofile> [-options]

Dubs one or more audio channels from a pure audio file into the audio portion of an existing clip. Currently supports .aiff and .wav. See the Operations section of the User Guide for details and examples.

### 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
```

---

**Print****Usage: *print***

Prints text following the command itself to the console. When used in a batch file, argument macros are expanded. The text to be printed must be surrounded by double-quotes. If no text is given, just prints a CR/LF.

---

**PullAud****Usage: *pullaud* <clip> <audiofile> [-options]**

Pulls one or more audio channels from a clip into an audio file. Currently supports AIFF and WAV formats. See the Operations section of the User Guide for details and examples.

**Options:**

-b -basechan n	First channel to pull
-c -chans n	Number of channels to pull
-d -delay n	has no effect
-t -trim n	has no effect
-s -start n	Start frame to pull
-f -frame n	Number of frames to pull
-w -wav	Output file will be WAV (AIFF default)

---

**PWD****Usage: *pwd***

Prints the current directory to the console. Basically an alias for the CD command with no path — included for UNIX-oriented users.

---

**QBOB****Usage: *qbob***

Contains the Qbob language compiler/interpreter to run QBOB programs. Usually called by the OS. Internal documentation available on request.

---

**Queue****Usage: *queue***

Kind of like showing the printer queue on a computer, except it shows clips that are playing or waiting to be played. Very techy. Useful for verifying if anything is “waiting to happen.” Does not impress people at parties, however.

---

**QQDownload****Usage: qqdownload**

QQdownload waits for data to arrive from the uploading QuBit and then restores the data locally to the current directory. No arguments or options are needed. Received data is written to the current directory, so be sure to cd to the correct download directory before invoking.

**QQUpload****Usage: qqupload <clip/file> <to IP address> [start xx] [frames xx]**

Begins sending data to the IP address specified. Currently, the “receiver” at that IP must be another QuBit running the QQDownload command.

**RecClip****Usage: recclip <name> [<frames>] [options]**

By default, records live video and 2 channels of audio from inputs 1 and 2 to the disk array. *Note: QuBit must be in record mode (see the vmode command).*

name        The name (and path) of the clip to create.  
frames      The number of frames to record before stopping. Specify 0 for indefinite (use the stopclip command or the STOP button on the QuBit front panel when you want to stop recording).

**Options:**

asyn        Command returns as soon as recording begins. Use STOPCLIP command to stop recording early. May not be used in conjunction with SYNC.  
sync        Command returns as soon as recording is complete. Press ctrl-c to stop recording early. May not be used in conjunction with ASYNC.  
sframe      Specifies single-frame recording. Each press of the REC transport button causes one more frame to be captured and stored. Press ctrl-c to stop before the specified number of frames has been grabbed.  
replace     Normally, RecClip stops with an error if the specified clip already exists. This option will cause an existing clip of the same name to be deleted, if it exists, without warning.  
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.  
noaudio     record video only.  
stats       provides console display of statistics during playback, and a summary. The statistics that are of interest to the user include Err (the number of dropped frames or other errors), TC (the current timecode), and Dat (the total amount of data consumed to that point).  
ltc|sys <hh:mm:ss:ff>   Selects the linear time code 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. Note that the timecode source that is selected to be recorded (via the set timecode, record command) does NOT have to be the same as the timecode that triggers the start of recording.  
hold        not documented.

---

**RecStat****Usage:** *recstat*

Displays information about the status of the previous `recclip` command. If recording is still underway, will return "Recording in progress." If recording has stopped for any reason (duration expired, ran out of free space, stop button pressed, `stopclip` command received), then the results of the last recording session are displayed as follows:

Last record result: 0 (Okay)

Last record size: 1800 frames

---

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

Attempts to recover data from the drive specified.

---

**Ren****Usage:** *ren* <oldname> <newname>

Renames a file or a directory from `oldname` to `newname`. Renaming a directory changes its name, even if it has files within it. `Ren` can also be used to move a file or directory (even if not empty) by specifying a different path. Notice that this does not work to move files or directories to another disk volume.

---

**Renvol****Usage:** *renvol* <name>

Renames the current volume to a new name. Notice that this is a drive name, as in "4.0," rather than a drive number, as in 0. The drive must be currently mounted and must have no files open on it for this command to succeed.

---

**SCSIList****Usage:** *SCSIList* [<channel> [<unit>]] [*VERBOSE*] [*CURRENT*]

Without arguments, `SCSIList` performs a SCSI update and makes any newly added SCSI devices available. Otherwise, you can specify which SCSI channel and unit number to add.

---

---

**SCSIrel**

**Usage:** *SCSIrel* <drive> <W|R|T> <blks> <DMAtype> [<passes>] [VERBOSE] [TERSE] [PAT x] [LBA x]

DMA type may be:

h HDMA

c CPU

b Bridge

f Far

hc or ch assymetrical test 'T'

Patterns can be a constant (0x1234), an inc (+0x110D), or TEST pattern.

---

**SerCtrl**

**Usage:** *ser* <port> <text>

Use this command to activate a serial port and have it translate one of the protocols listed below.

Supported protocols include QuBit, Odetics, LDV8000, BVW75.

Valid ports include SERc, SERd, SERa, SERb, 422a, 422b, DEBUG.

---

---

**Set**

**Usage:** *set* <group, item> <value>

*set* <group, item> <value> [save]

*set* <group, item> <value> restore

*set* <group, item> ?

*set* <group, item>

The set command will change environment variables (values that pertain to the broad group of settings that define QuBit's operating characteristics, see Appendix C for a list of environment variables and how you can use them to customize QuBit). Some changes are immediate (i.e., changing the number of audio channels you wish to record) while others require a reboot (i.e., changing the IP address).

Variables are divided into logical groups such as video, audio, network, etc. Each group contains a number of items related to that category.

(value)	Enters a new value for the specified variable. A temporary change only until the value is changed again, or QuBit is rebooted.
save	Saves the new value to non-volatile memory so that the it becomes part of the normal startup environment.
restore	Restores the "startup settings" in case they have been altered but not saved during this session.
?	Queries the usable values for a variable.
(no value)	To see the current setting, just type set followed by the variable group,item name.

---

**SetTC**

**Usage:** *settc* <hh:mm:ss:ff>

Sets the QuBit's time code clock to the value specified.

---

---

**Sharpen**

**Usage:** *sharpen* <n>

Qubit has the ability to (a) soften motion images while recording, (b) soften motion images during playback, and (c) sharpen motion images during playback. The sharpen command is used for any of these functions. It's best used symmetrically, so that the amount of softening applied matches the amount of sharpening applied.

To employ 9dB of softening when recording, issue the following command prior to recording a clip:

```
sharpen -3
```

-3	=	9dB softening	during record or on playback
-2	=	6dB softening	during record or on playback
-1	=	3dB softening	during record or on playback
0	=	off (default)	during record or on playback
1	=	3dB sharpening	playback only
2	=	6dB sharpening	playback only
3	=	9dB sharpening	playback only

---

**Shell**

**Usage:** *shell* <port>

This command opens up a shell on a specific QuBit port. Available ports include SERc, SERd, SERa, SERb, 422a, 422b, DEBUG. One use of this function is to open a second serial port during startup so that you can leave the LCD GUI running (which ties up serial port A) and still open up a terminal shell on serial port B for simultaneous GUI and serial shell control.

---

**Show**

**Usage:** *show* <group,item> | all

Use show to display the current status of a QuBit environment variable.

---

**Shuttle**

**Usage:** *playclip* <clipname> *shuttle*

*shuttle* <speed>

*shuttle stop*

The function of QuBit's digital shuttle enables you to shuttle through a clip. From the shell, you invoke shuttle on the playclip command line. Once in this virtual shuttle mode, you can enter new shuttle speeds without having to re-enter the entire line.

Use either *shuttle stop* or *stopclip* to exit shuttle mode and halt playback.

---

---

**Stat**

**Usage:** *Stat* [*<drive>* | *<dir>* | *<file>*]

Displays information about the specified drive, directory, file, or clip. This information, where applicable, includes datestamp, size, and permission bits. For clips, the duration in fields is also listed. For drives, details include the total size, percentage used, and amount free. If the drive(s) contain a fast and slow area, statistics will be listed for these separately. A fast area can contain video clips, whereas a slow area is only useful for non-realtime files, such as operating system files.

If no argument is entered, gives information about the current directory only.

---

**StopClip**

**Usage:** *stopclip*

Stops video playback or recording.

---

**Tape**

**Usage:** *tape*

The tape backup/restore program for QuBit. See the Operations section for instructions that walk you through archiving and restoring with

These commands affect the Exabyte Mammoth tape drive. They are a technical tool only and not supported. Where *< drv >* refers to the tape drive by its SCSI and unit numbers (see the end of Appendix L for a table of drive numbers).

Example (to eject a tape from the default tape drive SCSI ID via the shell): `tape 0x18 eject`  
See the chart at the end of Appendix J for valid SCSI ID combinations.

Options:

- rewind
- eject
- erase
- start
- stop
- marks *< num >*
- smarks *<num >*
- skip *< num>* *< type (0=blks,1=fm,4=sm,3=eod) >*
- locate *< lba>* *< part(-1=same) >*
- readpos
- blksize *< size >*
- getlog *< page >*

---

---

**Temp**

**Usage:** *temp*

*temp RANGE <low C> <high C>*

The first form displays the current ambient temperature inside the QuBit. The second form sets the safe operating temperature limits (low and high) in degrees C. If the internal temperature goes outside these limits, periodic warnings will be given (in the form of yellow or red icons on the LCD GUI).

---

**Unmount**

**Usage:** *unmount <drivenum>*

Un-mounts the drive specified. Notice that this is a drive number, as in 0, rather than a drive name, as in "4.0." The drive must be currently mounted and must have no files open on it for this command to succeed.

---

**Ver**

**Usage:** *ver [BOARDS] [CHIPS] [LIBS] [ALL]*

Displays selected version information for the system. Always displays at least the version of the software that's running and the Ethernet IP address of the unit. Additional information is also available when the following keywords are specified:

boards	lists type and revision of each hardware board installed.
chips	shows information about each programmable chip in the system.
libs	lists all software libraries loaded.
rls	lists details about the current release version of QuBit software
all	shows all the above.

---

**Vmode**

**Usage:** *Vmode [P | R | N]*

Controls the mode of the internal video software. P for play mode, R for record mode, and N for neither. If no mode is specified, will display the current mode.

---

**VSPMEM**

**Usage:** *Usage: vspmem [<test type>] [<test parts>] [Cx] [SWAP] [REPS <val>] [LEN <val>]  
[FILE <input>] [ONES] [LOG <log>] [ERRORS <val>|[ALL]] [ABORT] [FAST]*

Where: <test type> is one or more of the following:

ASRAM	Asynchronous SRAM tests. Valid Parts: EN
DRAM	DRAM tests. Valid Parts: EN QU PR
FIFO	FIFO tests. Valid Parts: PR XF
PASS	Pass through tests. Valid Parts: EN QU XF PR
	<test parts> zero or more of the following:
EN	Entropy encoder part.
QU	Quantifier part.
XF	Transform part.
PR	Predictor part.
MRG	Merge part.
HDMA	HDMA part.
DMA	DMA part.
MC	Master controller part.

Cx are channels to test (0-3). Default is all.

REPS <val> is the number of times total test is run.

LEN <val> is amount of data to test. Default is all.

FILE <input> is a file to get test data from.

SWAP performs byte swapping on test data (useful with FILE).

ONES appiles one's complement to test data.

LOG <log> is optional file for writing test results.

ERRORS <val> limits the number of errors that you will see. Default 20.

ERRORS ALL shows all errors in system.

ABORT exits program if max errors is reached. Otherwise the error counter is reset at end of each test and skips to next test if max errors is reached.

FAST Performs fast version of tests (if exists).

**Note:** To override the default VSP board, put use one of the following: [x,<test type>] where x is A-D. If no test part is given, it performs tests on all valid parts.

Default VSP Board is A.

**Wait**

---

**Usage:** *wait* <ms>

Causes a delay of the specified number of milliseconds. Most useful in batch files.

**WhatIs**

---

**Usage:** *whatIs* <error>

Prints text description of the specified error code.

**Which**

---

**Usage:** *which* <name>

Helps find a file that's being run at a command line.

For example, when you type "HD" and something happens, did it run a batch file called HD found in sys/bat, or did it run a module called HD in sys/mod, or a format file named HD in sys/fmt, etc.?

---

## Appendix I: Exabyte Mammoth Tape Drive Operations

### Mammoth I and Mammoth 2 (M2)

This appendix documents the Exabyte Mammoth and Mammoth 2 tape drive (commonly called M2).

---

#### ***Important! Use ONLY AME-Type Data Tapes!***

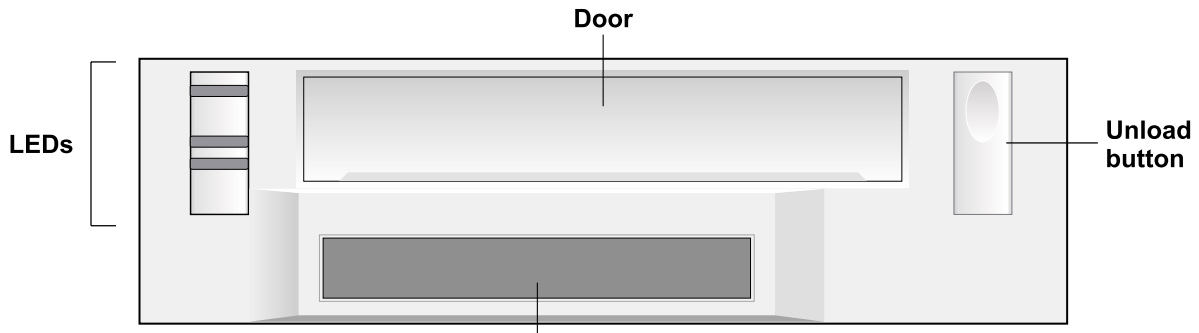
Although the Mammoth tape drive is manufactured by Exabyte, it does not support the use of older-format 8mm data tapes such as the popular Exabyte 8505. If one of these tapes is inserted, it can clog the tape heads. *Immediately* use the Exabyte Mammoth cleaning tape. Otherwise your QuBit will be unable to read any further tapes, even AME tapes. *Failure to follow this warning may void your warranty or result in costly downtime for repair.*

---

#### Operating the tape drive

This section describes how to operate the internal and external models of the tape drive. The figure below shows the front-panel components used for tape drive operation.

**Note:** Some models of the tape drive (such as the one included with QuBit) do not include the LCD panel



shown above.

#### Monitoring the LEDs

The LEDs have the following, general meanings:

- ❖ Top LED (amber).  
When this LED is flashing, an error has occurred.  
When this LED is on solid, the tape drive needs to be cleaned (see page 19).
- ❖ Middle LED (green).  
When this LED is on solid, a tape is loaded and the tape drive is ready to begin operations.
- ❖ Bottom LED (green).  
When this LED is flashing, tape motion is occurring.

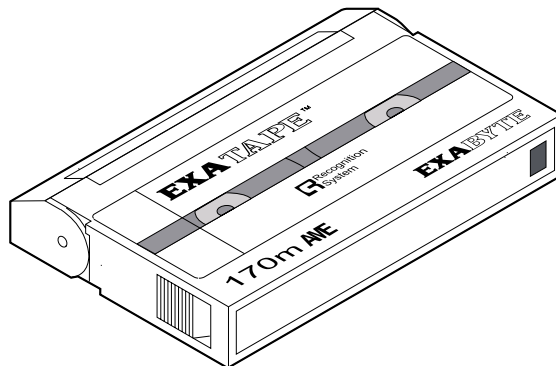
The following table describes LED combinations that occur during normal operation.

## Selecting appropriate data cartridges

	Tape Drive State							
	POST or reset	Error or failed POST	Ready (no tape loaded)	Ready (tape loaded)	Normal tape motion	High speed motion	Time to clean	Clean in progress
Top LED (Error/Clean)	●	*	n/a	n/a	n/a	n/a	●	●
Middle LED	(Tape Ready)	●	○	○	●	●	●	n/a
Bottom LED	(Tape Motion)	●	○	○	○	*	* fast	n/a

Legend: ○ = off   ● = on   \* = flash   n/a = not applicable (may be any state)

For writing data, use only EXATAPE™ advanced metal evaporated (AME) data cartridges in lengths of 22m or 170m only. *Other lengths are for other models of Mammoth drive and may not work with QuBit unless you have firmware version 41d or later. (To verify your firmware version, type “scsilist” at a telnet or serial shell. All SCSI devices and their versions will appear there.)*



The tape drive can also read data from metal particle (MP) tape when it is written in one of the following formats: 8500c, 8500, or 8200, but this is not recommended by QuVIS. (The tape drive cannot read data written in 8200c format.) After reading any MP tape, you **must** clean the drive with the cleaning tape.

---

### **Important!**

While Mammoth can read data written on metal particle tape, it cannot write to this type of tape.

---

### Loading data cartridges

To load a cartridge:

1. Make sure the tape drive is ready to accept a cartridge (all LEDs are off). Do not insert a cartridge if the tape drive is still performing its power-on self-test.
2. Set the write-protect switch for the desired operation.
3. Insert the cartridge.

The tape drive loads the tape in approximately 20 seconds. When the middle LED is on, the tape drive is ready for read and write operations.

### Unloading data cartridges

To unload a cartridge, press the unload button once to initiate the unload cycle. Do not press and hold the unload button for more than 10 seconds (this can cause a reset under certain conditions). If the tape drive is free of errors, it performs the following actions in about one minute:

- ❖ Completes any command in process
- ❖ Writes any buffered information to tape
- ❖ Rewinds the tape to the beginning
- ❖ Unloads the tape and ejects the cartridge

*Note:* If an error occurs before or during the unload procedure, the tape drive suspends the unload sequence. To clear the error, press the unload button again. The tape drive reattempts the unload sequence, but does not write data in the buffer.

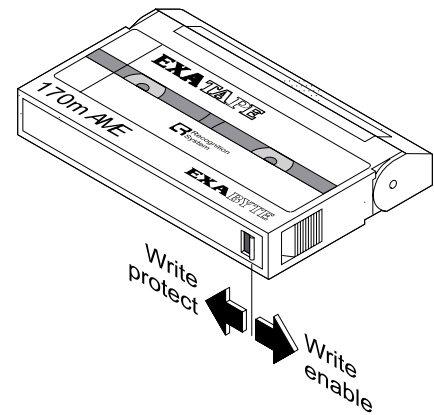
### Resetting the tape drive

To reset the tape drive, press and hold the unload button for at least 10 seconds. If this does not appear to be effective, you may prefer to power QuBit off and back on again.

*Note:* If you reset the tape drive while a cartridge is loaded, it rewinds the tape to the beginning after the reset is complete. The reset may take as long as two minutes if the tape is positioned near the end.

### Ejecting a Tape Manually

If for some reason you cannot eject a tape using the front panel button, you can issue a command to do so via a telnet or serial shell. The shell command `tapeutil` allows you to do this, as long as you specify the correct SCSI ID of the tape drive as part of the command. For example, using the default ID of the tape drive as shipped from the factory, the command to eject a tape is `tapeutil 0x18 eject`. See the SCSI ID chart at the end of Appendix J for valid SCSI device identification numbers.



### Cleaning the tape drive

When the tape drive requires cleaning, the top (amber) LED turns on. You should clean the tape drive as soon as possible after this LED turns on. The tape drive's cleaning requirements depend on the number of tape motion hours and the type of tape being used (MP or AME).

*Note: If an AME cartridge is ejected immediately after insertion, the drive needs cleaning.*

To clean the tape drive, insert an Exabyte Mammoth 8mm Cleaning Cartridge (or a cleaning cartridge approved by Exabyte for use with Mammoth). When finished, the tape drive turns off the top LED and ejects the cleaning cartridge.

### Updating the tape drive's firmware (internal code residing on-chip)

The firmware within the Mammoth tape drive can be updated with new data. If QuVIS releases updated code for the tape drive, you will be able to download the update and optionally install it.

### Maximum Data Capacities

Tape Length	Compressed Data	Uncompressed Data
22m	5 gigabytes	2.5 gigabytes
170m	40 gigabytes	20 gigabytes

## Troubleshooting

### Excessive Rewinding

Situation: During a backup or restore operation the tape mechanism suddenly sounds like it has changed to a fast-forward or rewind cycle. The tape heads are growing clogged and the drive is attempting to clear them.

Recommended Solution: Once the operation is completed, use a cleaning tape to clean the heads.

### Excessive Backup or Restore Time

Situation: Backup/restore operations take much, much longer than usual. The tape media is probably damaged or in otherwise poor condition. The drive is performing additional attempts to read each block of data before moving on to the next block.

Recommended solution: Once the tape is restored, back up the data to a fresh tape.

---

## Appendix J: Backup/Restore - The Tape Utility

The Tape utility is used to create and manage archives of QuBit files and clips on tape. All tape-oriented functions including backup, restore, verify, list, delete, etc. are performed using this one utility.

### The QuBit Data Format

The format of the data on a QuBit archive tape is a proprietary format (i.e. it is not TAR nor any other industry standard). A tape can hold one or more archives, limited only by the available space on the tape. Each archive is a collection of a number of files and/or clips. The items in an archive can be from various places on a QuBit's drives. There is no special meaning applied to the particular collection of items in an archive, it is just "a collection of things" stored in a single file on the tape. An archive is just a convenient way to group certain files or clips together. Perhaps the archive reflects the data on a machine on a certain date. Or perhaps all the files in an archive have some special relationship which makes it convenient to keep them bundled together.

### Formatting Tapes

QuBit archive tapes must be formatted before they can be used. This format operation can take a few extra minutes to complete. For that reason, the ability to erase a pre-formatted tape is also provided. Once a tape has been QuBit-formatted, it may be erased for reuse much more quickly by doing an erase operation versus a format. Either the erase and format operations can be done as a standalone operation, or can be done in conjunction with placing the first archive on the tape. See the Operations portion of the manual for details on formatting tapes.

### General Usage Overview and Notes

The usage of the Tape utility is simple: you enter the command followed by an operation and a set of options that specify additional details. If you ever forget the commands and options, just type `tape` at the shell and QuBit will display online help there. Here's an example template:

```
tape <operation> [<options>]
```

...where `<operation>` is one of the following (see below for details on each). Note that items enclosed in `<>` are *required*, whereas items enclosed in `[<>]` are *optional* and may be left out.

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

Here are some examples, whose options are explained further on.

```
tape list
tape backup * myarchive -v -q
tape restore lucas* sw @boot:clips\sw
```

In addition, the following [`<options>`] may be used in any combination:

Option	Abbrev	Description
-subs	-s	Descend into subdirectories
-query	-q	Manually confirm files to backup/restore
-clips	-c	Backup only clips
-files	-f	Backup only files
-verify	-v	Verify files after backup
-erase	-e	Quickly erase all archives on tape before backup
-format	-x	Format tape before backup
-unit <n>	-u	Force use of specific tape drive 'n'
-comp	-p	Compress data
-2nd	-2	Use second tape drive

The options may appear in any order, and anywhere on the command line. Except for the `-unit` option, the abbreviated forms can be grouped together (e.g. `-sqf`) when entered on the command line.

### Wildcard Support

Many of the operations support wildcard patterns where the filename or archive name are specified. A `?` matches any one character, while `*` matches any number of characters. Any number of these characters can be used in a pattern.

For example, given the following files on the disk:

Name	Type
NewDock1	clip
NewDock2	clip
New_Ship_final	clip
New_dock_final	clip
Dock.cfg	file

The following wildcard patterns would match the specified set of files:

Pattern	Matching Files
<code>new*</code>	NewDock1, NewDock2, New_Ship_final, New_dock_final
<code>n*final</code>	New_Ship_final, New_dock_final
<code>n*sh*final</code>	New_Ship_final
<code>*dock?</code>	NewDock1, NewDock2
<code>*dock*</code>	NewDock1, NewDock2, Dock.cfg

Patterns are not case-sensitive.

### -clips or -files

Further narrowing can be accomplished by also using the `-clips` or `-files` options. By default, all items that match the pattern given are accepted. If `-clips` or `-files` is specified, only the type of item specified will be accepted. Using `-clips` in the final pattern above, for example, would have prevented the use of the `Dock.cfg` file, even though it matches the pattern.

**-query**

The `-query` option can be used as a way to give the user absolutely the final say on whether to process a particularly named item. This can be used with any combination of patterns and/or `-clips` or `-files` filters. Any item that makes it through all other criteria will be presented to the user, who must accept or reject it by pressing Y or N.

**File patterns**

File patterns can also contain paths to indicate where to get files to be backed up. For example:

```
clips/john/Bik*
```

will only use files in the `clips/john` directory that start with "bik." This path is relative to the current directory at the time the Tape command is invoked, unless an absolute path is specified (e.g. `4.0/clips/john`). If no path is provided on a pattern, the current directory is assumed.

**-subs**

The `-subs` option can also be included to indicate that all directories should be searched, from the specified directory (or current directory) downward.

**The SCSIlist Command**

A useful command for verifying the presence of the tape drive, the SCSI ID of a device, or for obtaining the version number of the firmware on the tape drive is `scsilist`.

`SCSIlist` is particularly useful if you add another tape drive externally, or if you wish the correct SCSI ID of a device added to the SCSI chain. Type `scsilist` at the shell and press **ENTER** to force a refresh of the SCSI bus and to get information about the SCSI devices currently active on QuBit.

`SCSIlist` will also report the version number of the firmware on the tape drive. This is useful for verifying if your firmware code is recent enough to accept newer AME-type Exatape lengths.

See the chart at the end of this appendix for a list of SCSI chain and SCSI ID combinations.

### Tape List Command

#### *Example: tape list*

This operation simply provides a list of each archive found on the tape. The format of the list is shown below. Note that LIST reports the uncompressed archive size for each individual archive. Archives are listed showing the “on QuBit” size of an archive (the uncompressed archive as it would reside on QuBit), not the compressed “on the tape” size of the archive. At the end of the report you will find the amount of space used on the tape by all archives present, both compressed and uncompressed.

Archive	Serial#	Count	Size	Datestamp
TestArch	99962	6	29.9 MB	10-28-1997 15:33:29
UpdateStuff	99941	15	4.3 GB	12-14-1996 10:01:23

4.4 GB used, 14.2 GB left (24% full)

This example shows two archives, named TestArch and UpdateStuff. The rest of the data shows, for each archive listed, the serial number of the QuBit that wrote it, how many files it contains, the total data size, and the date and time it was created. For details about the contents of any or all of the archives, use the DIR operation.

The line below the list shows how much data is present on the tape currently, and how much estimated space remains on the tape.

*Note: All values are approximations. Slight inaccuracies can cause the amount of free space actually left on a tape to be slightly more or less than the amount reported.*

**Tape Dir Command****Example: *tape dir***

This operation shows the contents of one or more of the archives on a tape. It also provides the uncompressed size of the archive should you wish to restore it to QuBit. (See the UPDATE paragraph below for details about compressed data and the DIR command.)

The format of the command is:

```
Tape dir <archive>
```

A particular archive can be displayed, or all archives, as in the following examples:

```
Tape dir UpdateStuff
Tape dir *
```

Every archive whose name matches the pattern given will be displayed along with its contents. The format of these listings is as follows:

```
*** Archive 'TestArch' 24.2 MB Default path: 4.0/
Name                Type      Size      Status      Datestamp
sys/bat/ntsc.bat    File      98        Okay        10-24-1997 09:56:51
noam                Clip      00:00:00:06 Okay        10-23-1997 15:26:46
clips/ntsc          Clip      00:00:01:00 Okay        09-12-1997 23:59:20
4.4 GB used, 14.2 GB left (24% full)
```

The Type column indicates whether each item is a file or a clip. For files, the size column indicates the size of the file, in bytes. For clips, it indicates the length of the clip (expressed in timecode — HH:SS:MM:FF). The Status column indicates whether the item indicated 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 backup). Datestamp is the datestamp of the original file or clip.

The line below the list shows how much data is present on the tape currently, and how much estimated space remains on the tape.

**Compressed archives accessed from a tape backup**

The DIR operation provides uncompressed information about an archive file in case you wish to restore it to QuBit. Uncompressed data is shown for each archive on tape since this represents the “true” amount of hard disk space required to restore the archive on QuBit.

A sample line of the output of the DIR command may read as:

```
*** Archive 'Comp2' 8 GB Default path: 4.0/clips/
```

This demonstrates that the archive Comp2 will require 800 MB of disk space on QuBit, representing the original archive size in its uncompressed state. Data shown at the very end of the DIR report, where QuBit reports the total used space and free space remaining on the tape, reflects the total amount of information stored on the tape (comprising all archives, compressed or not). Here, the report shows the size of the archive(s) residing on the tape:

```
4 GB used, 14.6 GB left (23% full)
```

This is the actual data area that is used on the tape, plus the amount of free space remaining. For the example used here, the compressed archive Comp2 takes up 400 GB of space on the tape, while the uncompressed archive of Comp2 would require 8 GB when restored to QuBit. You can see that the file was compressed at 2:1 when it was written to the tape.

*Note: All values are approximations. Slight inaccuracies can cause the amount of free space actually left on a tape to be slightly more or less than the amount reported.*

---

**Tape Delete Command**

**Example:** *tape delete <archive name>*

This operation allows one or more archives on the tape to be deleted. The format of the command is:

```
Tape delete <archive>
```

The only limitation is that the archives to be deleted must be the last ones on the tape. In other words, when you delete archives, all archives from that point to the end of the tape are lost. The Tape utility automatically knows this and will prevent you from accidentally losing archives that you didn't mean to delete.

This operation is especially handy for deleting the last archive made. For example, maybe the wrong files were backed up, or maybe the archive failed or was aborted mid-backup. Just delete the archive and do it again.

---

---

**Tape Format Command**

**Example:** *tape format <archive name>*

This operation prepares a tape for use by the QuBit Tape utility. Specifically, it creates the necessary table of contents and data partitions, as well as other control information that is used to identify the tape.

This can be done using the `FORMAT` operation, which formats the tape and stops. It may also be done as a part of a backup operation by including the `-format` option. In this case, the tape is formatted, and then the backup operation begins.

The format operation is irreversible, and so a warning is presented to which the user must acknowledge that any previous data on the tape will be destroyed.

---

---

**Tape Erase Command**

**Example:** *tape erase <archive name>*

This operation quickly empties all archives from a tape. This cannot be done on a tape which has never been formatted for use on a QuBit.

This can be done using the ERASE operation, which erases the archives and stops. It may also be done as a part of a backup operation by including the -erase option. In this case, the tape is erased, and then the backup operation begins.

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.

---

### Tape Backup Command

---

**Example:** *tape backup <archive name>*

The format of the backup operation is:

```
Tape backup <pattern> <archive name> <options>
```

The pattern specifies which files and/or clips to backup. Archive name specifies the name to give the collection of files on the tape. Note that you cannot use an archive name that already exists on the current tape. To include spaces in an archive name, enclose the entire name in double-quotes, as in

```
Tape backup Noam* "Noam Set 12-23"
```

The `-subs`, `-clips`, `-files`, and `-query` options can be included to control, along with the pattern specified, which files and/or clips to backup.

By default, the tape drive's compression algorithm is not enabled. Addition of the `-comp` option will turn this on. Note that, depending on the data being backed up, the tape drive's native compression can often slow down an operation, and can even cause an increase in data size. This is especially true on data that has already been compressed by some means. This option may not be available on all makes of tape drives.

During backup, if compression is turned on, the effective compression ratio is reported at the completion of each file, as in the example that follows (which states that this file is 100% complete, and was compressed at roughly 1.5 times). This data is only shown during the backup process, on a file by file basis:

```
100% Compression ratio = 1.542:1
```

Other operations can precede the backup operation, including `-format` or `-erase`.

Note that the tape drive utilizes archival data compression and not motion image compression. All tape backup systems have built-in data compression to allow greater storage capacity on tape. In the same manner that a "pkzip" file is compressed/restored to its original data form on a PC, this form of compression does not affect image quality.

Specify `-verify` to do a verification pass on all files after the backup is complete. See the verify operation below for details.

The progression of a backup operation is as follows:

(Optional) format or erase operation is performed

List of files/clips to backup is processed, using supplied pattern and filters. May require user interaction if `-query` is specified.

Each matching file or clip is backed up

Final results and file list are added to the tape's Table of Contents

(Optional) verify pass is performed

---

At any point during the backup or during the processing that occurs before the backup actually begins, the operation can be aborted with Ctrl-C. If this is done, the integrity of the tape and archive are preserved. If no files have actually been written to tape yet, the result is that the tape is left exactly as it was before the operation was started. If one or more files have been (successfully or unsuccessfully) written already, then the archive is saved "as is" in case some of the data saved is usable. If this is not desired, the archive can be subsequently deleted. Any partial file that is aborted mid-backup is not accessible.

The above holds true if a fatal error occurs during backup which causes the backup to fail and cease – the integrity of the tape and archive are preserved and the archive is saved "as is." Simple failures of individual files do not cause the backup to stop; the offending file is just marked as "failed" in the table of contents. This allows the user to decide what to do with an archive that partially succeeds. It may be kept (and only the good files accessed) or it may be deleted.

## Tape Verify Command

---

**Example:** *tape verify <archive name>*

The verify operation compares the files in an archive to the originals 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.

One way this can be done is as part of a backup operation (by including the `-verify` option). In this case, all files and clips successfully backed up will be subsequently read back and compared against the original files.

This operation can also be started as a stand-alone operation. In this case, however, it's a little 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. The format of this operation is:

```
Tape verify [<pattern>] <archive name> [@<destination path>] <options>
```

If pattern is omitted, "\*" is assumed. Even in this case, any optional filters still apply. 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. Other subsequent archives are not verified (even if they match the pattern).

If no @destination path is included, the files are all verified against the files at their original location. The inclusion of a @destination path overrides this default, and specifies where to look for files to verify against the files on tape. The original structure of the archive is still preserved, but is transposed from the default path to the specified alternate @destination path.

When the verify operation is complete, a final report is given which summarizes the number of verification failures.

## Tape Restore Command

**Example:** `tape restore <archive name>`

The restore operation reads the files from the specified tape archive and writes them to disk. The format of this operation is:

```
Tape restore [<pattern>] <archive name> [@<destination path>] <options>
```

The four valid forms of the command are:

```
Tape restore <archive name>
Tape restore <archive name> @<destination path>
Tape restore <pattern> <archive name>
Tape restore <pattern> <archive name> @<destination path>
```

**Important:** *Be aware that the flexible nature of the tape restore command line may return a misleading error message if you incorrectly enter the command (which can result in a “no matching archives found” message). We recommend that you get in the habit of always using the pattern option, and -q option. To restore all files from an archive, as in examples 1 and 2, include the pattern “\*” for the pattern in the command line. To restore specific files from an archive, as in examples 3 and 4, then use an appropriate pattern to restore those files from the command line.*

**Archive name** is the archive to be restored. If wildcards are present in the archive name, then the first archive that matches the pattern is the one to be restored. Other subsequent archives are not restored (even if they match the pattern).

The **pattern** (and any of the filter options -clips, -files, and -query) specifies which files from the archive should be restored. If pattern is omitted, “\*” is assumed. Even in this case, any optional filters still apply.

The destination path each file in an archive is preserved from the original QuBit. If the destination is not otherwise specified on the command line, clips will be restored to the exact path from which they came. For example, if a file called RogersClip is backed up, and its original full path was 4.0/Clips/PreRel/RogersClip, then that is exactly where it will be restored to any other QuBit. If, however, @destination path is specified, the files are restored to disk using the path specified. You must use the “@” sign to specify an alternate destination path. The necessary sub-directories will be created if they do not already exist.

When restoring a file that already exists on your drive, you will be asked whether you want the restored file to replace the existing file or not.

**Troubleshooting Note:** *If during a restore operation the tape mechanism suddenly sounds like it has changed to a fast-forward or rewind cycle, the tape heads are growing clogged and the drive is attempting to clear them. Once the restore is completed, use a cleaning tape to clean the heads.*

## Query Option

When the `-query` option is specified, files that match the specified pattern and are not filtered out by the use of `-clips` or `-files` are presented to the user for approval. For each, press `y` or `n`. You may also press `X` to skip all future queries, or `A` to accept all future queries.

*Note: To abort an operation entirely from a query prompt, you must press `Ctrl-C`, then press `Enter`.*

## Unit Option

The `-unit` option, followed by a number, specifies the SCSI channel & ID of the tape drive to use for backup. If omitted, the first tape drive found is used. An easy way to specify another tape drive (other than the first) is to use the `-2` option, which ignores the first drive found and uses the next higher SCSI drive available.

The format for specifying the tape drive to be used is this: the prefix “0x” followed by the number of the SCSI chain that the drive is connected to, followed by the Hex number that represents the number of the SCSI ID the drive has. It’s all written as one “word.”

Consult the table below for the correct unit number to use on the command line.

	SCSI chain 1	SCSI chain 2	SCSI chain 3	SCSI chain 4
SCSI ID 0	<b>0x10 *</b>	<b>0x20 *</b>	<b>0x30 *</b>	<b>0x40 *</b>
SCSI ID 1	0x11	0x21	0x31	0x41
SCSI ID 2	0x12	0x22	0x32	0x42
SCSI ID 3	0x13	0x23	0x33	0x43
SCSI ID 4	0x14	0x24	0x34	0x44
SCSI ID 5	0x15	0x25	0x35	0x45
SCSI ID 6	<b>0x16 **</b>	0x26	0x36	0x46
SCSI ID 7	0x17	0x27	0x37	0x47
SCSI ID 8	<b>0x18 ***</b>	0x28	0x38	0x48
SCSI ID 9	0x19	0x29	0x39	0x49
SCSI ID 10	0x1a	0x2a	0x3a	0x4a
SCSI ID 11	0x1b	0x2b	0x3b	0x4b
SCSI ID 12	0x1c	0x2c	0x3c	0x4c
SCSI ID 13	0x1d	0x2d	0x3d	0x4d
SCSI ID 14	0x1e	0x2e	0x3e	0x4e
SCSI ID 15	0x1f	0x2f	0x3f	0x4f

\* *These first four IDs are the default factory-installed drives in QuBit.*

\*\* *This is the default ID of the DVD drive, if factory-installed.*

\*\*\* *This is the default ID of the Exabyte tape drive, if factory-installed.*

## Appendix K: Play List Creation

A play list is a text file (with a .lst extension) that you run from either the LCD GUI or a shell. The file's contents include commands that specify a method of display (i.e., play, or still) and the name of the clip to be displayed. Playlists may have up to 200 entries (lines).

Here's a short overview of the process:

- Create a text file on your computer, save it with a .lst extension (as in playlist.lst)
- Use an FTP program to transfer the file to QuBit (Appendix X: "How to Transfer Files to QuBit")
- Run the file from the GUI (select the file, then press the *green triangle button* labeled Queue) or from the shell (type `playlist` followed by the name of the play list file).

### Example Play List

This example includes samples of all play list options, and is written as if it were a playlist itself.

---

; The first line of every playlist must begin with a semicolon, which indicates that it is a "comment." A  
; comment serves as a placeholder where you may give the playlist a title, or write notes to yourself or oth-  
; ers who may wish to edit it later.

; There are five playlist commands: PLAY, LOOP, STILL, SMPTE, and WAIT. They may be used freely  
; within any script (although the SMPTE command is best used at the beginning only).

; The PLAY command is written in the form of

; PLAY clipname inframe durationframes

; as in this example

```
PLAY intro 0 1000000
```

; or this example using a relative pathname

```
PLAY /trailers/intro 15 600
```

; or this example using an absolute pathname

```
PLAY boot:/clips/intro 30 9000
```

; As a shortcut, if you do not know the last frame of a clip, you can use a very high number as shown in  
; the example above. The clip will play to its end and, since there are no further frames, QuBit moves on to  
; the next clip. Otherwise, it's best to use frame numbers if you know them.

;

; If the script itself resides in the same directory where the clips it refers to are located, then nothing more  
; is needed than the name of the clip. If the clips reside in a separate directory, apart from the script, then  
; you will need to enter either the relative path or the absolute path to them. Relative pathnames assume the  
; current dir is the one containing the script, and other paths are written in relation to this starting location.  
; Absolute pathnames always begin at the root directory of QuBit.

; The LOOP command is written in the form of

; LOOP number

; as in this example

```
LOOP 5
```

; Which will repeat the entire playlist five times before stopping

---

; The STILL command is written in the form of

; STILL clipname inframe durationframes

; as in this example

```
STILL intro 0 90
```

; or this example

```
STILL boot:/clips/intro 30 90
```

; QuBit thinks of all motion imagery as clips whether they are still images or motion images. The still command allows you to display a still, or a still frame within a clip, for a duration of time. In the playlist, you should specify the name of the clip, the frame to display, and the length of time (in frames) that it should be displayed.

; As with the PLAY command, you may specify clips by their name only, by their relation to the current directory, or by their absolute pathname.

; The SMPTE command is written in the form of

; SMPTE hh:mm:ss:ff

; as in this example

```
SMPTE
```

; or this example

```
SMPTE 03:45:26:15
```

; The SMPTE command performs one of two functions, it either: sets the time code value at the beginning of playback to the value specified; or begins each clip in the playlist at 00:00:00:00 if no value is specified.

; The WAIT command is written in the form of

; WAIT durationframes

; as in this example

```
WAIT 2
```

; or this example

```
WAIT 30
```

; The WAIT command causes QuBit to pause during playback. The pause value is written in frames.

---

## Appendix L: Customizing QuBit's Startup and Operations

When QuBit is powered on, a series of commands is issued that set up QuBit for operation. You can edit certain startup files to modify QuBit for your environment and work habits, or for specific playback situations such as a synchronized multi-screen multi-QuBit presentation. The commands available to you are those that set environment variables (see Appendix C “Environment Variables”), those that run commands or software modules that function like internal commands (Appendix H “QuBit Shell Commands”).

### The User Directory

Within the QuBit directory structure you will find a directory named *user* at the root level, with a subdirectory named *bat*. This directory contains files for each registered user who operates QuBit.

*Note: If you use the security features of QuBit's operating system, then there may be a selection of subdirectories, one for each registered user. The instructions for modifying general user settings files in this appendix apply equally to the settings files for specific users.*

### The Custom.bat File

Custom.bat is the last file consulted during startup. It is the preferred place to put global custom or machine-specific settings, and it's useful for setting the proper image size and operating mode. It's also a good place to put batch files that provide aliases for your most often-used video modes.

Here's an example of a custom.bat file:

```
    ; $Id: custom.bat,v 1.9 1999-10-25 02:40:13-05 Stratton Exp $
    ; Add user specific customizations to this file only below this point
    print "User Customization File:CUSTOM.BAT"
    print ""
    hd2_1280x1026_24p
    vmode p
    set video,lumsnr 54
    ; Add user specific customizations to this file only Above this point
    IF EXISTS BOARD(UIF)
        print ""
        print "Starting Graphical User Interface"
        print ""
        print "Type <ctrl>+D to exit GUI"
        print ""
        GUI
    ENDIF
```

This batch file sets QuBit to start up at an image size of 1280 by 1026 with a display rate of 24 frames per second, sets up for playback, sets the user's preferred recording SNR to 54 dB, and then runs the GUI on the front panel.

Note that there is a specified “customization section” pointed out in the commented second line. This is intended to make modifications and troubleshooting simpler by grouping setup changes together into one section of the file. Place any modifications there, and restart QuBit for them to be active.

### The Userplay.bat File

The file `userplay.bat` is used to customize your preferences for playback options (options that cannot be set from the front panel). It runs immediately after QuBit is set to playback mode (that is, after the `vmode p` command has been issued).

For example, you may prefer to have QuBit “hold” on the final frame of playback rather than go to black (this is particularly useful for still frame presentations, where you want the viewers to view a sequence of slides). To do this, you would insert a line to set the environment variable *hold* to the value *frame*.

Here's an example of the file `userplay.bat`:

```
; $Id: userplay.bat,v 1.2 1999-10-24 23:44:58-05 Stratton Exp $
; Add here any changes or setup needed when going into PLAY modes
set video,hold frame
```

This file tells QuBit to hold the final frame of any playback for an indefinite period of time (until stop is pressed or another clip is played). This is particularly useful for presentations, where you want the viewers to view a sequence of slides.

*Note: Userplay.bat is the last file accessed by QuBit after you press the PLAY button but before playback begins. Therefore, the settings in this file override **any** settings on the GUI.*

### The Userrecord.bat File

The file `userrecord.bat` is used to customize your preferences for recording options that cannot be set from the front panel. It runs immediately after QuBit is set to record mode—that is, after the `vmode r` command has been issued.

Here's an example of the file `userrecord.bat`:

```
; ; $Id: userrecord.bat,v 1.3 1999-10-24 23:43:49-05 Stratton Exp $
; Add here any changes or setup needed when going into RECORD modes
; Set skew
skew vio_parin +2
```

This particular example sets the skew value for recording.

*Note: Userrecord.bat is the last file accessed by QuBit after you press the RECORD button, and before recording begins. Therefore, settings in this file will override any conflicting settings on the GUI (for example, you could use this file to “lock” QuBit so that it could only record in a specific image size and at a specific SNR, no matter what the GUI settings showed.)*

## Appendix M: Mirroring Hard Drives for Critical-Reliability Playback Applications

### What's Mirroring, and Why Mirror?

Traditional mirroring in the computer industry allows you to maintain an exact duplicate of the data on one hard drive on a second hard drive. The purpose is to provide playback confidence by having an online (or near-online) backup at hand should the original drive fail for any reason.

When switched on, mirroring enables two exact copies of hard drive data to be “online.” This means that if any hard drive reports a problem during playback, QuBit can invisibly switch to a backup drive without any audio or visual interruption of the program. This is particularly useful for theatrical presentations, but is applicable literally anywhere uninterrupted playback is important.

With QuBit mirroring is a bit different. With mirroring active, QuBit understands there is a second hard drive set that contains mirrored motion image data, which will be used if needed. Unless the mirror data is created specifically by use of the `copydata` command, or by setup during a digital cinema DVD install, QuBit does not automatically update the mirror drive set. Should you write a new script file or other data to QuBit manually by ftp, know that it is not copied to the mirror set automatically. This note is neither a problem nor a drawback; it is presented for your information.

### Requirements

- Mirror drives must be the same brand, model number, and size as the original drive set.
- The drives should be internally mounted using 68-pin SCSI cables that have extra inline to accommodate multiple drives on a single SCSI chain.
- Mirror drives should have SCSI ID 1. The original drive set is set to SCSI ID 0.

### Mirroring A Drive Set

With the primary and mirror drives properly set up, boot QuBit. Open a shell and issue the command `scsilist`. You may need to issue it a second time. Should any drive not appear in the drive list, check your cabling, reboot, and type `scsilist` again until QuBit displays all eight drives.

Once all drives are present, issue the command `mirrordrives` from a shell. It may take up to several hours, depending on the size of the drive set being mirrored. The QShell will display the percentage of completion throughout the process until done.

*Note: The command `mirrordrives` is actually a batch text file, `mirrordrives.bat`, that copies the main drives to the mirror drives. Any existing data on the mirror drives will be overwritten by this.*

### Mirroring a single file (or multiple files)

To mirror a single file from the primary drive set to the mirror drive set, issue the command `mirror (filename)` press Enter. Here, `(filename)` is the name of the file to be mirrored. This is best used when you wish to copy a single file from one mirror set to another, perhaps an updated script file.

This will copy the specific file to the backup set. Be aware that the mirror command writes data from the primary set to the mirror set in mirror-fashion, duplicating the clip on the mirror set at the same drive location as that where the original clip existed. This is not a simple copy operation where data is written into the next available block of space. It is duplicated block for block.

**Auto mirror switching**

If a set of mirror drives is detected, playback will automatically switch to the mirror drive(s) as needed when an unrecoverable error occurs on one drive. QuBit does not currently attempt to switch back to the original drive set until an error occurs on the mirror drive set.

The CineGUI shows the total number of such errors, and an "(m)" indicator to show when we're using the mirror drives.

**Additional notes**

When you want to mirror a primary drive set to a secondary drive set, use the `mirrordrives` batch command. Enter `mirrordrives` from a command shell.

When you want to update the mirror drive set with a file that's been changed on the primary set, use the `mirror` command. Enter `mirror filename` from a command shell.

To turn mirroring on so that QuBit will utilize the mirror drive set during playback, use the environment variable `set scsi,mirror true`. Place this command in the `custom.bat` startup file so that whenever drives are detected at startup, mirroring will be enabled. The state `false` is set automatically if no mirror drives are found, so it is safe to leave this `true` command in place in your `custom.bat` startup file.

To turn mirroring off use the environment variable `set scsi,mirror false`.

---

## Appendix N: Supported Edit VTR Commands

QuBit supports RS-422 edit controllers using a set of QuVIS-implemented VTR commands.

To use an edit controller with QuBit, you will need to:

- (a) Hook up QuBit to the edit controller via a 9-pin serial cable (wired straight-through).
- (b) Power on both pieces of equipment.
- (c) Make sure all QuBit clips are of the same image size and mode.
- (d) Make sure all QuBit clips are in the same directory. By default, this directory is root:clips/bvw.
- (e) Start the protocol on QuBit. Issue this command via a shell:

```
serctrl bw75 on 422a
```

Or use 422b for the second serial port. Once the protocol is started up on the QuBit, it cannot be stopped until a reset. Optionally, you can have this command issued as part of the custom.bat startup file so that QuBit is automatically “listening” upon startup – see Appendix L “Customizing QuBit Startup”.

As currently implemented, QuBit supports the following commands in various capacities.

CMD	Function	Returns	Result
00.11	Device Type Request	0x1211	0xE022
20.00	Stop	ACK	Stops the clip
20.01	Play	ACK	Plays the clip
20.04	Standby off	ACK	Does nothing
20.05	Standby on	ACK	Does nothing
20.0F	Eject	ACK	Does nothing
20.10	Fast Fwd	ACK	Advances the clip at 100x speed
2X.11	Jog Fwd	ACK	Advances the clip based on data given
2X.12	Var Fwd	ACK	Advances the clip at the speed given
2X.13	Shuttle Fwd	ACK	Advances the clip at the speed given (QuBit does not differentiate between this command and Var Fwd)
20.20	Rewind	ACK	Backward at 100x speed
2X.21	Jog Rev	ACK	Reverses the clip based on the data given

2X.22	Var Rev	ACK	Reverses the clip at the speed given
2X.23	Shuttle Rev	ACK	Reverses the clip at the speed given. (QuBit does not differentiate between this command and Var Rev)
20.30	Preroll	ACK	Positions clip at previously set timecode minus preroll length
24.31	Cue up with data	ACK	Positions clip at given timecode
40.10	In entry	ACK	Saves current timecode
40.11	Out entry	ACK	Saves current timecode
41.30	Edit Preset	ACK	Does nothing
44.31	Preroll time preset	ACK	Sets preroll time to given data
41.36	Timer mode select	ACK	Sets timer mode to given value
61.0C	Current time sense	74.04	LTC time data (current clip's time)
61.20	Status sense	7X.20	The requested status data
60.36	Timer mode sense	71.36	Time mode data

**Note:** *These can be changed with the command "set protocol,bvwclip <name of new clip>".*

## Appendix O: QuBit Startup and Diagnostics

During startup you can monitor QuBit's status if you have a serial shell open prior to power up (see the Installation section for details on the serial hookup).

One option you have during startup is to press the spacebar of your computer or terminal to interrupt the process and optionally run a set of diagnostic functions. If you wish to do so, here are the optional startup choices that you may toggle on or off from the screen.

- |                    |  |
|--------------------|--|
| (F) Full SCSI Scan | Performs a scan of the entire SCSI bus and then updates QuBit's internal ROM with the result. Updating the ROM tells QuBit what SCSI devices are currently connected, and that they should be mounted automatically upon every future startup. This process requires about 2-4 minutes.<br><br>This is different from the <i>scsilist</i> command. <i>SCSIlIst</i> will only update the SCSI bus for the current session. Once QuBit has been powered off, any devices found by <i>SCSIlIst</i> will no longer be available. |
| (R) ROM Boot       | Boots the QuBit from its internal ROM rather than the hard drive. This is primarily a troubleshooting and technical support function.  |
| (S) Safe Boot      | Another technical support function. Starts up the QuBit from disk with only minimal functionality and support.   |
| (M) Memory Test    | Performs a thorough memory test of the RAM within QuBit. Takes about 5 minutes.  |
| (C) Continue       | Once you have toggled one or more options above, press C to initiate the diagnostics.  |



## Appendix P: CineGUI (Preliminary Documentation)

*Note: CineGUI and its operations are part of a worldwide pilot program for digital cinema. As such, these documents may change at any time. See [support.quvis.com](http://support.quvis.com) for updated versions.*

CineGUI is a graphic interface for the QuBit in digital cinema operations (theatrical playback-only environments). It is designed to be easily understood and operated by non-technical personnel. CineGUI allows the theater owner/operator the ability to load new material (“content” in the form of trailers, films, and more) from DVD discs, and start any feature film using a simple control screen.

CineGUI allows production studios the ability to generate uninterrupted show scripts from multiple reels using a powerful yet easy-to-read-and-understand scripting language. It also features four GPI trigger outputs that can be used to trigger theatre lighting, curtains, etc., in an automated theatre.

Following is an overview of CineGUI operation and Qscript creation.

### From the Studio to the Theater

The current “distribution model” for digital cinema is this: feature films and trailers are transferred to a mastering QuBit, divided into DVD-sized pieces, written to DVD-R discs, duplicated, and then distributed to remote theatre locations. The exact processes and software are not explained in detail here. Complete documentation of the process can be obtained by contacting the Sales and Marketing Department of QuVIS, Inc. Other forms of distribution including Internet, broadband, and satellite are currently under investigation.

### Loading a Feature Film from DVD onto QuBit

Begin loading by inserting the first DVD of the set into QuBit's DVD reader. When a valid QuBit DVD disc is inserted into the DVD drive, it is automatically detected and QuBit will ask if you would like to load this disc set.

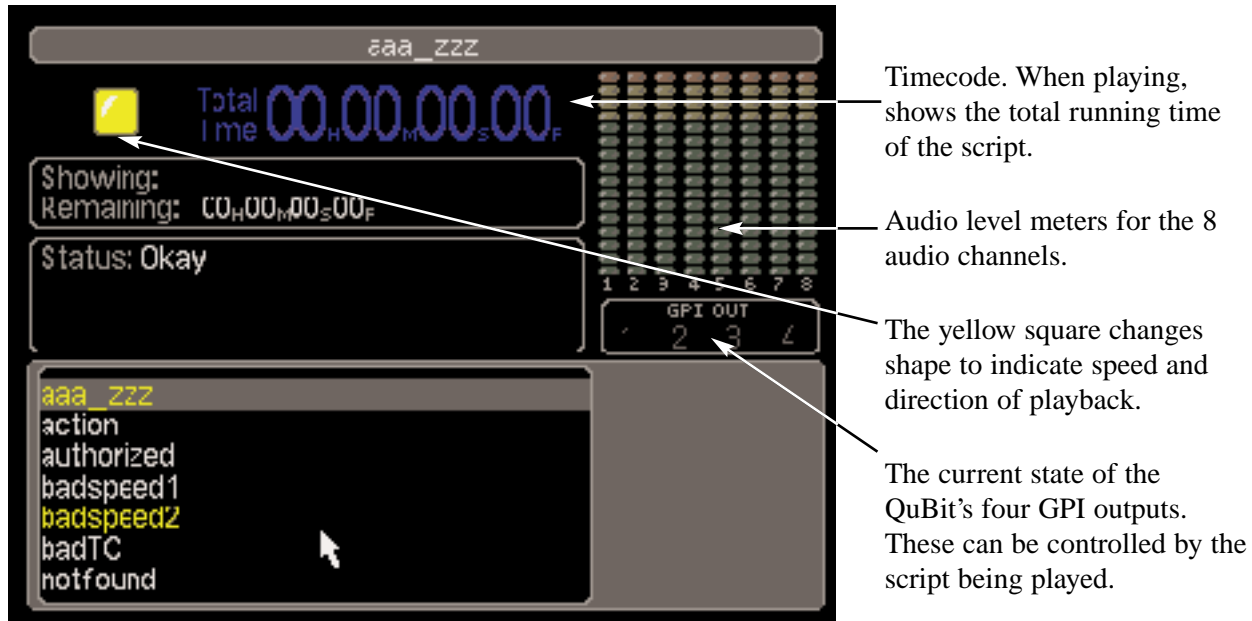


If security is enabled for the disc, you will be required to enter a password before the content can be loaded to the QuBit. Enter the password key supplied with the discs by clicking on the appropriate letters and numbers provided. If a keyboard is attached, you may use it to enter the key more easily.

At this point the contents of the disc will be loaded. When QuBit is done with the current disc, it will request the next disc in the set. You may not load the discs in the incorrect order. Once the last disc has been loaded, the clips will be indexed and mirrored, and the process is complete.

## Main Display

Following is a short discussion of the main display and its features:



Timecode. When playing, shows the total running time of the script.

Audio level meters for the 8 audio channels.

The yellow square changes shape to indicate speed and direction of playback.

The current state of the QuBit's four GPI outputs. These can be controlled by the script being played.

Above three windows: The name of current clip in progress (playback), along with its time remaining; the status window, showing any warnings of errors detected; finally, the script window (the list of scripts available on this QuBit). Click on one to place its name in the title bar, which signifies that this script is selected for playback.

*Note: Scripts are found in the /scripts directory.*

## Transport Controls - How to Play a Script (Movie, Presentation, Etc.)

### To Play a Script

Select the script from the touchpad and press the PLAY button. If QuBit has been paused on a still frame, it will resume playback from that point. From here, you can STOP or SCAN through the script.

### To Stop a Script

Hold STOP and press BACK. (Similar to GPI input #2, panic stop.)

### To Pause a Script

Press STOP. Pauses the script and displays a still frame. From here you can PLAY or STEP through the current script.

### To scan through a script at high speed

Press the REV or FWD buttons while the script is playing.

### To step through the script a frame at a time

First, pause the script by pressing the STOP. Then press the REV or FWD button to advance by frame.

## Scripts

A script is a text file that contains instructions to tell QuBit how to play various segments of content without interruption (as if they were one seamless production). Once written, the script is placed in the QuBit's /scripts directory and CineGui is started (or the QuBit is rebooted). When CineGUI appears, scripts appears as menu item which may be selected and played.

To start a script, select it and press the PLAY button on the QuBit transport buttons (or close GPI input 1). During playback, you may pause the script indefinitely on the current frame by pressing the STOP button. The light on the STOP button will illuminate while the output is paused. To resume playback, press the PLAY button again.

**Note: GPI input 2 is a PANIC STOP input.** When closed, playback is immediately stopped, and the screen output goes black.

When the PLAY button is pressed, the selected script is first scanned for errors. If an error is found, it is displayed on screen and the offending line is displayed along with the line number at which it can be found in the script. If there are no errors in the script, playback immediately commences.

**Note: During script development and initial playback:** Only one error at a time is reported (the first one encountered). After the first error is corrected, and the script is again played, the next error is reported. It may take several play-and-correct cycles to completely fix a script for error-free playback.

### Script Format

The script format allows the scriptwriter the freedom to generate a master script that contains two distinct parts: a "feature" portion that contains the commands necessary to play the separate feature film segments as if they were one complete "virtual reel;" and a customizable "local" portion that is specially setup for the different requirements of each theater (including the theater chain's logo, trailers, and in-house GPI signals for lights, curtains, etc.).

In this manner a studio releasing a feature film—via DVD—can generate a single "feature" script without changing the "local" script and accidentally causing playback problems by overwriting previously-correct GPI data, for example.

The first line of a QuBit script must always contain the following comment starting in column 1:

```
:Qscript
```

After that, their format is rather free-form. Each step in the script is described on its own line. Also, blank lines and comment lines may be inserted with no effect.

The following command lines may be used in any order and combination to build the show as required:

```
;Line of comments

PLAY      <clipname>  <inpoint>      <outpoint>
STILL     <clipname>  <point>        <duration>
GPI       <chan1-4>   ON | OFF | PULSE <frm>      [NOW | -<time> | +<time>]
AUDIODELAY <frms>
GROUP     <path>
SCRIPT    <script_name>
SWITCH
DATE      <mm-dd-yyyy>                (only within SWITCH lines)
TIME      <hh:mm:ss>                  (only within SWITCH lines)
ID        <string>                    (only within SWITCH lines)
```

### PLAY command

Each PLAY line lists one file (or clip) to be played. The first parameter is the name of the clip (this may be either the full pathname to the clip, or just the name of the clip if it resides in the “/clips” directory, or if the GROUP parameter has been used—which specifies a path—just the name of the clip).

Next is the “in point” in the clip, specified in timecode notation (HH:MM:SS:FF). Following this, the next parameter is the “out point” of the clip, specified in the same manner. As a shortcut, you may use END as the outpoint. These two points specify the range of the material that should be played from within the specified clip.

This example plays “reel1” from the beginning of the clip to 19 minutes, 17 seconds, and 23 frames.

```
PLAY /Reel1 00:00:00:00 00:19:17:23
```

This example specifies the exact location of the clip. Note the word END for the out point, specifying the last frame in a clip.

```
PLAY /clips/extra/fix14 00:00:00:00 END
```

### STILL command

The STILL command displays a single frame from a clip, lasting for the specified duration. Like PLAY, STILL requires the name of the source clip. However, in this case the “in point” specifies which frame of the clip should be displayed and held. The last parameter is the duration for the event, describing how long to hold the still (in timecode).

This example plays frame 00:00:01:18 out of the “logo” clip and holds it for 2 seconds

```
STILL /clips/Logo 00:00:01:18 00:00:02:00
```

**GPI command**

Each GPI command line describes a signal to be generated on a particular GPI output.

There three parameters for the GPI command:

The first parameter specifies *which output* is being controlled. This should be a number from 1 to 4.

The second parameter is the *output type*: A hard ON, a hard OFF, or a timed PULSE (on-off, which also requires you to specify the pulse width, in frames; use a simple number, not a time code notation).

The third parameter describes *at what time* the event should occur, or start to occur in the case of a timed pulse. The default is to have the event start at the exact time in the script where the GPI command is placed, usually between two clips. You would either specify "NOW" or provide no final parameter for the event to occur at this point. Optionally, you may specify a start time that is before or after this instant in the script by specifying the time difference, preceded by a negative (-) or positive (+) sign. This allows you to control external events synchronous to certain key frames in a nearby clip, regardless of when that clip actually occurs in the script.

This example turns on output 1.

```
GPI    1    ON    NOW
```

This example creates a pulse 10 frames long on output 3.

```
GPI    3    PULSE 10    NOW
```

This example creates a pulse 30 frames long on output 4, five seconds before the previous clip ends.

```
PLAY   someclip           00:00:00:00    END
GPI    4    PULSE 30    -00:00:05:00
```

**AUDIODELAY command**

AUDIODELAY allows the QuBit's to advance or delay the audio signal in relation to the video playback. This is useful when clips have been recorded with the video out-of-sync with the audio, or to allow for video delays caused by routing QuBit's video output through a digital device with a framebuffer that delays the picture by several frames. It is a simple matter to have the QuBit adjust its internal delays to make each piece more consistent using the AUDIODELAY command.

The only parameter needed is the amount of delay, which is expressed in video frames. The allowable range is -7 (audio advanced ahead of video by 7 frames) to 23 (audio delayed from video by 23 frames).

The delay change occurs at the exact instant in the script where this command appears. You cannot insert a delay change in the middle of a clip unless the clip itself is played as if it were two separate clips with the delay change between. This would require two PLAY lines in the script, one for each portion of the clip, with the AUDIODELAY command line inserted between them.

This example advances the audio ahead by 2 frames (typical of framebuffer device routing):

```
AUDIODELAY          -2
```

### GROUP command

This command sets the default path to look for clips specified in subsequent **PLAY** and **STILL** commands. This is mostly an organizational tool to reduce the clutter in scripts. Multiple **GROUP** commands may exist in a script. The most recent one that precedes the **PLAY/STILL** command is the one in effect.

The path specified is always assumed to be inside the **/CLIPS** directory. For example, a script such as this would look for a clip named **"/clips/new/test/alpha."**

```
GROUP          new/test
PLAY          alpha 00:00:00:00  END
```

### LOOP command

This command allows you to repeat the entire script—either indefinitely or for a specific number of repetitions. **LOOP 0** repeats the entire script indefinitely, **LOOP 1** repeats it once, **LOOP 2** twice, etc. This is more suited for repeating presentations such as preview kiosks than for scripting theatrical playback. Note that you cannot loop individual clips (to do that, simply repeat the play command line as many times as you need).

### SCRIPT command

This command allows one script to be started by another script. This is useful for organizing a program into smaller mini-scripts that can be maintained by separate parties independently, yet still work together. Each script can be referenced from the "master" script using the **SCRIPT** command, which points to one or more other scripts to be run during the course of executing the master script. The primary benefit of this form of scripting is to allow a studio to release complete "feature" scripts that do not interfere with a theater's existing "localized" scripts.

For example, a theater may have one master script that calls three sub-scripts - the first being the local theater chain's logo, the second being upcoming trailers, and the third being the feature itself. Any one of the three sub-scripts, along with the directories that contain their script-related content, could be replaced with new material (new logo, new trailers, and/or new feature) from a DVD distribution without having to create a new master script. In this way, once a system is created, it need only be maintained.

See below for an example of a set of these "nested" scripts.

There is no limit to the "depth" to which you can nest scripts. It is improper, however, to reference a script that references the original script. In other words, if script A runs script B at some point, script B cannot run script A, as this would produce an endless loop. The **LOOP** command is a legal means of producing a script that will repeat forever.

You can make scripts invisible from the CineGUI, so that sub-scripts that are run from the master script will not appear. Name them starting with an underscore (e.g. **"\_trailers"**). Scripts named this way can still be run from other scripts, but cannot be run by the user since they are not "visible" from CineGUI.

**SWITCH command**

This command begins a sequence of optional script lines. Think of this as a detour that “branches” the script. Decisions can be written into the script that determine which script lines ones are followed and which ones are skipped. A SWITCH END line marks the end of this section and must always be present.

The format of a block of lines inside a SWITCH is as follows:

```
SWITCH
(0 or more script commands)
DATE/TIME/ID test
(0 or more script commands)
[0 or more DATE/TIME/ID tests and script commands]
SWITCH END
```

**Example 1a:** To switch script lines on a particular date, use the following:

```
SWITCH
;(script lines to run if date/time has not been reached go here)
DATE 11-25-1999
;(script lines to run if date/time has been reached go here)
SWITCH END
```

A real-world example in the above format would look like this:

```
SWITCH
PLAY creditroll_1 00:00:00:00 00:07:00:00
DATE 11-25-1999
PLAY creditroll_2 00:00:00:00 00:07:00:00
SWITCH END
```

**Example 2:** To switch at a precise time on that date:

```
SWITCH
;(script lines to run if date/time has not been reached go here)
DATE 11-25-1999 TIME 18:20:00
;(script lines to run if date/time has been reached go here)
SWITCH END
```

There can be many alternate sections. Just make sure the dates are in ascending order (last dates are the latest):

```
SWITCH
(script lines to run if none of the following dates reached)
DATE 11-25-1999 TIME 08:00:00
(script lines to run if 1st date/time has been reached)
DATE 12-10-1999 TIME 08:00:00
(script lines to run if 2nd date/time has been reached)
DATE 01-14-2000 TIME 08:00:00
(script lines to run if 3rd date/time has been reached)
SWITCH END
```

You can also list just times (no dates). This allows a script to be created which changes depending on the time of day, and does this the same way every day. This is useful for a theater that plays different films at different times on the same screen:

```
SWITCH
(script lines to run if time is 12:00 am to 9:59:59 am)
TIME 10:00:00
(script lines to run if time is 10:00 am to 4:59:59 pm)
TIME 17:00:00
(script lines to run if time is 5:00 pm to 11:59:59 pm)
SWITCH END
```

The following example makes a new clip just "show up" on a certain date (doesn't replace anything):

```
SWITCH
DATE 07-12-1998 TIME 11:00:00
PLAY newclip 00:00:00:00 END
SWITCH END
```

This example plays different clips depending on the UserID of the machine that's playing it:

```
SWITCH
ID EP008
PLAY watermark1 00:00:00:00 END
ID SP021
PLAY watermark2 00:00:00:00 END
ID SP053
PLAY watermark3 00:00:00:00 END
SWITCH END
```

**Notes:**

- 1) Lines in alternate sections do not need to be indented, but should be for clarity.
- 2) Time is expressed in 24-hour format
- 3) Time and dates in scripts refer to local time. Make sure QuBit's real-time clock is set correctly.

**Example: Show Script**

```
;Qscript

;Example theatrical script

;Dim lights
GPI          1          ON

;Play local theatre logo
PLAY          /clips/TheatreLogo  00:00:00:00      END

;Play trailers (this may change often)
PLAY          /clips/Trailer1     00:00:00:00      END
PLAY          /clips/Trailer2     00:00:00:00      END
PLAY          /clips/Trailer3     00:00:00:00      END

;Get ready for feature
PLAY          /clips/OurFeature    00:00:00:00      END

;Change sound system
GPI          2          ON

;Play the feature
AUDIODELAY   0
PLAY          /clips/reel1         00:00:00:12      00:18:38:13
AUDIODELAY   -1
PLAY          /clips/reel2         00:00:02:03      00:19:55:05
PLAY          /clips/reel3         00:00:01:07      00:21:08:21
AUDIODELAY   2
PLAY          /clips/reel4         00:00:00:19      00:20:20:19
AUDIODELAY   1
PLAY          /clips/reel5         00:00:03:21      00:18:45:13

;Bring house lights back up 2 minutes 18 seconds before feature ends (credits)
GPI          1          OFF          -00:02:18:00

;End of show (black)
```

### Example: Nested Scripting

Here's an example of a working set of three separate scripts. In the course of running the master script, the other two sub-scripts are also run. These two are the “nested” scripts, named with an underscore so that their names do not appear on the CineGUI screen; only the master script is visible. Note that this is only an example, assuming a single feature per QuBit. A different script naming convention and directory structure would have to be created to support multiple feature films on a single QuBit.

These three scripts could have been written as a single script. However, this would shift the burden of script-creation to the originating studio, which would have to create and maintain custom theatre-specific scripts for *every* screen in every theatre complex. Changes in equipment or operation of any theatre that were not known by the studio could mean that a script might be “broken” and not work.

The flexibility of nesting lets the studio create a single “feature” script that does not rely on any conditions other than the clips it refers to, and which are installed with the feature script. For this to work, each script has its own “private area” for content to allow different parties to control the different parts of the presentation without affecting the others, and without any one party needing to retrieve the “current script” to make modifications to one small part. Either the “previews” script (and its content area) or the “feature” script (and its content area) could be updated independently of one another without affecting the overall playback structure. In this way either the previews or the feature could be updated at will.

Note that the “extra spacing” between commands and their options on each line is shown in a tabbed, columnar fashion here for readability. There is no hard and fast rule about how many spaces should be placed between items on a line. It's useful for fast visual organization.

```

=====
;Qscript
;scriptname: theatre1
GROUP    local
GPI      1 PULSE 10          NOW          ;Lights partially down
PLAY     theaterlogo       00:00:00:00 END
SCRIPT   _trailers
GPI      2 PULSE 10          -00:00:01:00 ;Lights fully down for feature
PLAY     feature_presentation 00:00:00:00 END
SCRIPT   _feature
GPI      4 PULSE 10          -00:00:01:00 ;Bring lights up to full
=====

```

*Above: The theatre\_1 script starts by defining a path for the first clip to be played: /clips/local, sends a pulse to dim the house lights, then plays a clip named theatrelogo that resides within that directory.*

```

=====
;Qscript
;scriptname: _trailers
GROUP    trailers
PLAY     trailer1  00:00:00:00          END
PLAY     trailer2  00:00:00:00          00:00:03:29
PLAY     trailer3  00:00:08:01          00:00:04:12
=====

```

*Above: The \_trailers script follows. This script sets a different path for the content: /clips/trailers, then plays the previews.*

```

=====
;Qscript
; scriptname: _feature
GROUP      feature
PLAY       r1ab      00:00:00:00      00:20:00:00
PLAY       r2ab      00:00:00:00      00:20:00:00
PLAY       r3ab      00:00:00:00      00:20:00:00
GPI        3 PULSE 10                      ;Bring lights up for credits
PLAY       credits   00:00:02:00      00:05:09:29
=====
    
```

Above: The `_feature` script. This script sets a different path for the content: `/clips/feature`, then plays the feature film. Although the film is made up of 4 separate clips, they play back seamlessly and without interruption.

Here is another way of looking at this nested script set: listed event by event showing how the scripts handle each item in turn. This may help serve as a visual explanation of how the process flows.

	Theatre_1 script	_Previews script	_Feature script
event 001	Set content directory to /clips/local		
event 002	Send GPI pulse: dim lights		
event 003	Play theatrelogo clip		
event 004	Run <code>_previews</code> script		
event 005		Set content directory to /clips/trailers	
event 006		Play trailer1 clip	
event 007		Play trailer2 clip	
event 008		Play trailer3 clip	
event 009		(return to previous script)	
event 010	Send GPI pulse: lights off		
event 011	Run the <code>_feature</code> script		
event 012			Set content directory to /clips/feature
event 013			Play r1ab clip
event 014			Play r2ab clip
event 015			Play r3ab clip
event 016			Send GPI pulse: lights up to dim
event 017			Play credits clip
event 018			(return to previous script)
event 019	Send GPI pulse: lights up to full		
event 020	(end of program)		



## Appendix Q: Synchronizing Multi-QuBit Playback with Time Code

To control two or more QuBits from a common time code source, here are the engineer's notes from a recent installation.

### General

All QuBits should be running the same version of system software, should be hooked up to a common time code source, and have a common computer connection (Ethernet or serial) to act as a coordinating controller. One QuBit acts as the master; any additional QuBits act as slaves. The master QuBit parallel output connected via 68-pin SCSI cable to the slave, and daisy-chained via additional SCSI cables to any other slaves in the system.

The QuBits are given commands to play a specific clip, beginning on a specific frame, at a specific moment in time.

### Hooking Up the Equipment

1. Connect the controlling computer to the master QuBit (serial or Ethernet connection).
2. Connect the controlling computer to the slave QuBits via Ethernet.
3. Connect the LTC generator output to all QuBits. Use a Y cable (or distribution amp) to split the signal if necessary.
4. Open up a shell connection to all QuBits. Set them to the proper video mode, and put them in a playback state.
6. Connect the parallel digital output of the master QuBit to the parallel digital input of the slave.
7. On the slave QuBits, run the batch file `slave.bat`. Type `slave` and press Enter. It should be found in the `sys` directory. This establishes frame-lock between the two QuBits.

*Note: To verify frame-lock, disconnect the parallel input from the slave QuBit. It's video output should now display "snow." This breaks that QuBit's output, so you will need to reconnect the parallel input and run `master.bat`, followed by `slave.bat`, to restore and reset it to slave status.*

### Starting Synchronous Playback

8. Cue up each QuBit: enter the `playclip` command (as described below) to start at the proper frame and time. Each QuBit will ready the specified clip, then stand by for the time code trigger.

The command template for each QuBit to start playback based on time code is:

```
playclip <clipname> <inpoint> <# of frames>
LTC <desired start time>
```

So, as an example, you would enter:

```
playclip mainevent 27 2500 LTC 00:00:05:00
```

This will instruct QuBit to wait for 5 seconds before playing the clip called *mainevent*, starting at frame 27, for a duration of 2500 frames.

*Note: You do not have to use an inpoint and a duration. They are optional command switches that we used here as an example.*

9. Start the time code generator. Using the example command line from above, at 5 seconds each QuBit should begin playback. In your situation, at the trigger moment, each QuBit should begin.

### Resetting for Next Playback

10. Reset the LTC generator to 00:00:00:00. (Some generators may need their power cycled to do this.)
11. Reset each QuBit by re-entering the "playclip" command line and pressing Enter (shortcut: you can press the up arrow on the keyboard to recall the last-issued `playclip` command, and then press Enter). Do this for each QuBit.



## Appendix R: Recording and Playback with Serial/Parallel HD Converters

If your QuBit is not equipped with “live” serial digital connections and you plan to use serial signal sources with QuBit, you may need to use a serial-to-parallel converter (for input/recording) or a parallel-to-serial converter (for output/playback).

Below are some notes and details you may wish to consult. See also Appendix E for the pinout diagram of QuBit's parallel connectors, should you wish to build a custom cable.

### Recording using a [Serial-to-Parallel] HD Converter

QuVIS has used YEM and Miranda serial-to-parallel conversion devices with general success.

---

***Important! YEM Converters must be configured properly***

Conversion equipment manufactured by YEM is designed to be changed in the field to adhere to the pinout specifications of several different manufacturers. It is possible to connect a parallel cable with an incorrectly configured YEM and damage your equipment. Always make sure you have the correct cabling and pinout configuration before connecting and operating your equipment.

---

The general hookup procedure is simple. Connect the serial digital output from your playback device to the serial digital input of the converter. Then connect the serial-to-parallel converter's parallel digital output to QuBit's parallel digital input.

*Note: QuBit's parallel digital input is a 68-pin D-sub connector. You may need to create a custom cable for this connection. See Appendix E for pinouts.*

### Playback using a [Serial-to-Parallel] HD Converter

QuVIS has used YEM and Miranda serial-to-parallel conversion devices with general success.

---

***Important! YEM Converters must be configured properly***

See the note about YEM equipment above.

---

The general hookup procedure is simple. Connect QuBit's parallel digital output to the input of your converter. Then connect the parallel-to-serial converter's serial digital output to your recording (or display) device.

*Note: QuBit's parallel digital input is a 68-pin D-sub connector. You may need to create a custom cable for this connection. See Appendix E for pinouts.*



## Appendix S: QuBit Security Features

Depending how accessible QuBit is to untrained personnel, you may wish to establish greater security to protect vital recordings from deletion or corruption.

To limit the capabilities of other people who use QuBit, you can set up QuBit with a system of “users” who have “privileges,” or access to specific functions. The administrator of user/privilege status assigns these privileges, which allow your facility to exercise a degree of security over who operates QuBit and what they can do with it.

### Privileges and Passwords

By default, QuBit ships with its security features disabled, which allows the user to customize the system freely. However, you may wish to enable “QuBit privileges” which are assigned to known QuBit operators and which may be used to limit access to sensitive commands.

Currently there are three levels of privileges, each allowing a different degree of operator freedom. Privileges are honored from all QuBit interfaces (shell, telnet, and ftp).

### Access levels supported:

- User - No access to sys/security directory. Many commands disallowed.
- Technician - No access to sys/security directory. Most operations supported.
- Supervisor - Full access to sys/security directory. All operations supported. Administrators receive supervisor status.

### To Set Up Users and Privileges

To establish a set of users and their privileges, you must edit a file called *users* that resides on QuBit (although this is a text file, it does not have the standard “.txt” extension as a further security measure). To edit the file you first transfer the file from QuBit to a computer. Then you can modify it using a standard text editor (like Notepad on Windows). See Appendix X, “How to Transfer Files to/from QuBit” if you are not sure how to do this.

Here is an example *users* file:

```
;USER PASSWORD PRIVELEGE (user, technician, supervisor)
Marty xyzyy supervisor
Dave tygyop technician
Dana tune77 user
```

Only users with SUPERVISOR access can read, delete, or write the *users* file.

**Note:** *If this file does not exist for any reason, there would be no way for a user to obtain SUPERVISOR status. Therefore, there is a default user named SETUP with the password PLEASE, that will give supervisor status until a /SYS/SECURITY/USERS file exists. Once this file exists, the user SETUP and the password PLEASE no longer function.*

### Shell User Security Functions

By default, when you open a serial or a telnet shell, you will begin operations under “user” status. Use the commands *login* and *logout* to change your status.

For example:

```
login Marty XYZZY
```

If this matches one of the entries in the *users* file, then the privileges are updated to the level listed in the file *for that shell only*. If this doesn't match any entries, the shell is returned to “user” status.

To restore “user” status to the shell, use the *logout* command.

For example:

```
logout
```

### FTP User Security Functions

FTP programs usually feature a connection or options panel with fields where you can input the user name and password. If there is no security set up for QuBit, the standard logins of either Anonymous or Guest with a password in the form of *guest@domain.com* will serve.

However, once security is active on QuBit, you can only connect via FTP if you use a login and password that exist within the *users* file. Your supervisor may wish to add an “anonymous” user with a specific password for just such cases.

*Note: A special case is user named "QTools" who gets TECHNICIAN status.*

## Appendix T: Creating Batch Text Files to Run from the GUI

You can create self-running “batch files” which may be initiated from the standard front panel GUI with the **QUEUE** button. The benefit of this is that you can create self-running demonstrations, for example. You can automate almost any process that can be otherwise run from a command line.

In one sense, this is a form of playlisting or scripting. However, playlists and scripts are limited to a handful of specific commands that they “know” such as play, still, and loop. Batch file creation draws upon the entire QOS (QuBit Operating System) shell command set (Appendix H).

For example, a recent demonstration of digital cinema required clips of different formats to be played while guest speakers discussed various aspects of digital cinema content creation, distribution, and operations. The clips had been recorded in different resolutions, requiring that QuBit be “set” for the correct image mode before each clip could be run. Complicating matters, security had been setup for this QuBit.

A set of batch files was written to issue this sequence of commands: login password (with security active, an authorized login is required to change modes), set the image mode, set for playback, change to the correct directory, and then play the clip with a specified in point and duration. By hand, issuing these commands would not have been difficult; however, with a few hundred important guests present, you don't want to make a mistake “on the air.” Therefore, a set of batch files was written and tested beforehand.

For the presentation, the operator needed only to select the batch file name from the GUI screen and click the **QUEUE** button to start the playback.

These files could also have been run from a shell, just by typing their name (without the .bat extension).

### Requirements

In order for batch files to work, they must be written as standard ASCII text files and have the extension “.bat,” as in *autoscript.bat*. They need to contain standard QShell commands along with the required options in order to run fully. Finally, they may reside any directory; however, to be available from the command line at any time and from any directory, we recommend *root:user/bat* as the location for your own customized batch files.

To play these customized scripts from a shell, simply enter the name of the file *minus the .bat extension*. If the files reside elsewhere, you will need to `cd` to that directory first before you can run the file.

To run them from the GUI, navigate to the File/Clip Manager screen then navigate the directory structure of QuBit to *root:user/bat*, select the file, and click on the **QUEUE** button.

Alternately, if you plan to run these from the GUI only, you could simply place them directly in the /clips directory, so that they can be accessed more quickly. Just remember this location should you later wish to run them from a shell.



## Appendix U: Creating Uncompressed D-1 Format Single-Frame Animations

If you wish to create a D-1 format (NTSC or PAL) animation in an uncompressed, 8-bit format, you can do so by transferring a sequence of stills to QuBit's *magic* directory.

### **Using an FTP client (All OS's)**

When you transfer still images to the *magic* directory on QuBit, they are converted to a clip. Images must be either ### x ### (NTSC) or ### x ### (PAL). Images transferred to the *magic* directory are not compressed with QuBit's highly-efficient, data-saving techniques, and will therefore require more disk space than a compressed clip containing the same material.

*Note: This procedure assumes you have already setup QuBit for Ethernet/FTP transfers as described in the installation section of the manual.*

- 1 Open your FTP client and establish a connection to QuBit using its IP Address (in the same manner that you establish an FTP connection with a remote computer over the Internet, for example).

If you're not sure of QuBit's IP Address, check it by either looking at the Setup menu on the LCD GUI or by typing *ver* at a telnet or DOS shell.

- 2 Navigate to both the source directory on your local machine and the destination directory on QuBit (this will be the *magic* directory).

Most FTP clients allow you to display both a local directory (on your computer) and a remote directory (in this case, on the QuBit).

- 3 Select the files to be transferred and begin the transfer.

As files arrive in the *magic* directory, they are processed into a growing clip file and then deleted (as they are no longer needed.)

- 4 Once the process is complete, you may use the GUI and play the newly created clip. Or you may play it via a telnet or serial shell.



## Appendix V: LD-V8000 Laser Disc Protocol

### LD-V8000 Setup

This is a quick primer on how to setup for LD-V8000 test:

1) Connect cable from QuBit serial port B to whatever device will be controlling QuBit.

2) From shell, start the serial control protocol:

```
SerCtrl LDV8000 on SERb
```

3) LD-V8000 does not understand the concept of clips, so you must tell the QuBit which one clip to use for the virtual laserdisc. Do this like so:

```
set protocol,LDVclip boot:clips\TexasWild
```

If the clip name is specified without a path like this, it is assumed to be in the /LDV directory. Always specify full, absolute pathnames starting from the root of the QuBit (e.g. boot:clips/test/TexasWild).

4) Make sure QuBit is in proper play mode for the clip selected.

5) QuBit is now ready to respond to LD-V8000 protocol. Note that we currently only support 9600 baud, 8-N-1 for communications. This is adjustable on real LD-V8000 players.

Notice that you can "load a different disc" (a different clip) by changing the *protocol,LDVclip* variable, but it will not take effect until you reject the current disc (RJ) and spin-up again (SA). (Technically, the door should be opened (OP) and then closed again (CO), but that seemed excessive.) You can also load different clips using the QuBit-specific extended command LU (see below).

### Supported Commands

The following commands are supported in the QuBit implementation of LD-V8000. Those in italics are accepted but non-functional.

```
DOOR OPEN .....OP
REJECT .....RJ
START .....SA
PLAY .....(address) PL
PAUSE .....PA
STILL .....ST
STEP FORWARD .....SF
STEP REVERSE .....SR
SCAN FORWARD .....NF
SCAN REVERSE .....NR
MULTI-SPEED FORWARD .....(address) MF
MULTI-SPEED REVERSE .....(address) MR
SPEED .....argument SP
SEARCH .....address SE
STOP MARKER .....address SM
FRAME .....FR
TIME .....TM
CHAPTER .....CH
CLEAR .....CL
FRAME NUMBER REQUEST .....?F
```



**Appendix W: Placeholder (no Appendix at this time)**



## Appendix X: How to Transfer Files to/from QuBit

With an Ethernet connection to QuBit and FTP client software on your computer you can transfer files to and from QuBit. This tutorial, geared toward transferring a play list to QuBit but applicable to any type of file transfer, assumes you've already hooked up to QuBit and verified an Ethernet connection.

The specifics for each FTP client software program vary, however the basic steps are fundamental to all programs. Use your judgement as needed to follow these steps.

### Where to Find an FTP Program?

FTP programs are available all over the Internet, from your local computer software reseller, and may be built-in to your computer operating system. Most any FTP program will work for this process.

- 1 Start your FTP program.
- 2 Check the "options" or "preferences" settings of your ftp program and make sure that it is set for "non-binary" file transfers.

*Note: You can keep this connection open to QuBit at the same time as a telnet shell. The convenience of this is that you can continue to fine-tune the play list, send it to QuBit, then play it, as many times as you wish without having to open and close each program as you work.*

Binary file transfers (either to or from QuBit) will usually create files that cannot be used.

- 3 Open the connection panel to enter the IP Address of the QuBit you wish to connect to. Enter the address and click the button to make the connection.
- 4 Navigate to the appropriate local directory (on your computer).

Most FTP software uses the concept of a "local" and a "remote" directory. These serve as the source and destination directories for file transfers.

- 5 Navigate to the appropriate remote directory (on QuBit) .
- 6 Select the file(s) you wish to transfer (whether they are on the QuBit or on the computer).
- 7 Press the "transfer" button to send the files to their destination.



**Appendix Y: Placeholder (no Appendix at this time)**



## Appendix Z: Document History

The pages that follow serve as a history of the changes that have been made at each stage of documentation. Consult this list with each new revision to learn “what’s new.” Latest changes will be at the bottom.

(c) 2000 QuVIS, Inc.

this file: appendices.history.txt

references file: appendices[date].qxd (appendix files for QuBit)

maintained by: jhebert@quvis.com

contents: contains descriptions of updates/change made to QuVIS product documentation.

4.17.2000

saved as appendices[04.17.2000].qxd

created appendices[04.17.2000].pdf

4.18.2000

updating playlist functions (appendix K)

saved as appendices[04.18.2000].qxd

4.19.2000

added sharpen to shell commands

saved as appendices[04.19.2000].qxd

5.02.2000

updated several environment variables

merged shell commands and software modules into one alphabetical list

added hddiags mod to shell commands

saved as appendices[05.02.2000].qxd

5.03.2000

fixed error in tape drive appendix

fixed some formatting

updated CineGUI (appendix P)

saved as appendices[05.03.2000].qxd

5.15.2000

added a tip for ejecting the backup tape via the shell (appendix I)

added some info about default SCSI IDs to SCSI ID table (appendix J)

saved as appendices[05.15.2000].qxd

highlight: with system release 1.59 DVD-loading is now up to full drive speed - twice as fast as before.

05.22.2000

added LOG command

typofix: HDDiag had the heading "Help"

typofix: Shuttle had a comment from it sister function Jog.

updated JOG command usage

cleaned up DIR command a bit

added appendix F: updating QuBit OS software

5.23.2000

Minor pathname change.

5.24.2000

Corrected misalignment in graphic.  
Revised Appendix P playlisting. Much more thorough examples.

5.25.2000

Added note for DACTEST TRIM.

6.08.2000

Added notes to Exabyte and tape appendices about head clogging during restore.

6.19.2000

Updated HDDiag command.

7.03.2000

Updated tape drive appendix

7.18.2000

Updated the Ver command

7.20.2000

Updated the tape drive appendix with notes about "wrong length" tapes. Customer had tried to use a 125m length tape that is designed for Mammoth LT drives only.

7.21.2000

Added to Appendix I:  
More changes to "Selecting Appropriate Data Cartridges." Turns out that 125m length tapes work with more recent firmware only.  
Added section on Ejecting Tapes Manually.

Added to Appendix J:  
Some SCSIlist notes.

Added to Appendix H:  
Note on tape eject via the shell, added to the Tape command.

7.31.2000

Appendix I updates for Mammoth Tape Drive  
Began revising CineGUI docs.

8.02.2000

Revised text throughout Appendix P: CineGUI.

8.07.2000

Incorporated "History" files into documentation (always the final page).

8.08.2000

Revisions/updates to Appendix Q - Synchronizing Multi-QuBit playback.

8.22.2000

Added LOOP command to CineGUI scripting.

8.29.2000

Updated Appendix B network troubleshooting and settings.  
Updated "inputsrc" environment variable to include SerialSH and SerialHD choices (VI08).

9.14.2000

Updated Appendix N: Mirroring drives (mirrordrives batch file and mirror file command). Also updated opening paragraphs of Appendix C: Environment variables.

---

09.15.2000

Added support contact info to front page

09.18.2000

Major revision of first two pages of Appendix C: Environment variables, with examples.

09.29.2000

Revised Appendix A with additional examples of serial port troubleshooting.

10.02.2000

Correct typos throughout (thanks to a silent contributor).

Revised Appendix M with additional information.

Updated Mirror shell command.

