

VCP-Recorder User's Guide

Document Number 007-3620-001

CONTRIBUTORS

Written by Susan Patick

Illustrated by Dany Galgani

Production by Allen Clardy

Engineering contributions by Rick Reed, David Worthington and Mike Moskowitz
St. Peter's Basilica image courtesy of ENEL SpA and InfoByte SpA. Disk Thrower
image courtesy of Xavier Berenguer, Animatica.

© 1998, Silicon Graphics, Inc.— All Rights Reserved

The contents of this document may not be copied or duplicated in any form, in whole
or in part, without the prior written permission of Silicon Graphics, Inc.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure of the technical data contained in this document by
the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the
Rights in Technical Data and Computer Software clause at DFARS 52.227-7013
and/or in similar or successor clauses in the FAR, or in the DOD or NASA FAR
Supplement. Unpublished rights reserved under the Copyright Laws of the United
States. Contractor/manufacture is Silicon Graphics, Inc., 2011 N. Shoreline Blvd.,
Mountain View, CA 94043-1389.

Silicon Graphics and the Silicon Graphics logo are registered trademarks, and IRIX,
Extent File System (EFS), XFS, Origin, Origin200, Origin2000, Onyx2, and O2 are
trademarks of Silicon Graphics, Inc. UNIX is a registered trademark in the United
States and other countries, licensed exclusively through X/Open, Ltd.

Contents

List of Figures xi

List of Tables xiii

About This Guide xv

What This Document Contains xv

Who Should Read This Document xvi

Related Documentation xvi

Conventions Used in This Document xvii

- 1. Overview of VCP-Recorder** 1
 - Functional Overview 1
 - Software Overview 4
 - Clip Cache 6
 - Core Software 6
 - Archive Interface Module 7
 - Control Interface Module 7
 - Storage Device Interface Module 7
 - Format Interface Module 8
 - Media Device Interface Module 8
 - Logical Playback and Record Units 9
 - Hardware Overview 11
 - VCP-Recorder Server 13
 - Origin Servers 13
 - O2 Workstations 13
 - Disk Storage 14
 - External Devices 14

- 2. Using the VCP-Recorder GUI 15**
 - Playing and Recording Clips 16
 - Starting the VCP-Recorder Media Control Panel 16
 - About the Media Control Panel 22
 - Playing or Recording an Existing Clip 29
 - Creating a New Clip 31
 - Changing Cue Points and Edit Points 32
 - Controlling a Video Deck 33
 - Accessing the Deck Control Window 33
 - About the Deck Control Window 34
 - Recording From the Deck to a Clip 37
 - Recording From a Clip to the Deck 38
 - Monitoring the Status of VCP-Recorder 39
- 3. Using the Clip Manager 41**
 - Starting the Clip Manager 42
 - Clip Manager Menus 44
 - Obtaining Information About a Clip 45
 - Renaming a Clip 46
 - Deleting a Clip 47
 - Setting the Protections for a Clip 48
 - Using VCP-Recorder With an Archive System 49
 - Locating a Clip in an Attached Archive System 50
 - Bringing In a New Clip From an Attached Archive System 51
 - Updating a Clip 52
 - Writing a Clip to an Attached Archive System 53
- 4. MVCP Reference 55**
 - Using the MVCP Commands 55
 - Establishing an Interactive MVCP Connection 56
 - Rules for Using MVCP Commands 56
 - About MVCP Timecodes 57

| | |
|--|----|
| MVCP Responses | 59 |
| 2xx Response Codes | 60 |
| 4xx Response Codes | 61 |
| 5xx Response Codes | 62 |
| Synchronization of Unit Commands and Responses | 62 |
| Changing the Synchronization Mode | 63 |
| Changing Command Sequencing | 64 |
| Command Timing | 65 |
| MVCP Global Commands | 66 |
| AFND (Find Clip in Archive) | 68 |
| AFNG (Find Archive Get) | 69 |
| AGET (Get From Archive) | 70 |
| ALSG (List Archive Gets) | 71 |
| ALSP (List Archive Puts) | 72 |
| APUT (Put to Archive) | 73 |
| ARM (Delete From Archive) | 74 |
| ARMG (Cancel Archive Gets) | 74 |
| BYE | 74 |
| CADD (Add Clip) | 75 |
| CCHP (Change Clip Protection) | 76 |
| CCP (Copy Clip) | 77 |
| CCST (Clip Cache Status) | 77 |
| CEDP (Set Edit Point) | 78 |
| CGP (Get Clip Protection) | 78 |
| CIMG (Create Clip Image) | 79 |
| CINF (Clip Info) | 80 |
| CLN (Link Clip) | 81 |
| CLS (List Clips) | 82 |
| CLSA (List Added Clips) | 83 |
| CLSR (List Removed Clips) | 83 |
| CMIN (Clip Monitor Info) | 84 |
| CMON (Clip Monitor) | 84 |
| CMV (Rename Clip) | 85 |

| | |
|------------------------------|-----|
| CRM (Delete Clip) | 85 |
| CRMA (Delete All Clips) | 85 |
| ERR (Error) | 86 |
| FRAT (Set Frame Rate) | 86 |
| GTOD (Get System Time) | 87 |
| MON (Monitor) | 87 |
| PLS (List Ports) | 91 |
| SORD (Set Sort Order) | 92 |
| STLS (List Statistics) | 92 |
| STOD (Set System Time) | 93 |
| STST (Statistics Statistics) | 93 |
| STZ (Reset Statistics) | 94 |
| UADD (Add Unit) | 95 |
| ULS (List Units) | 96 |
| MVCP Unit Commands | 97 |
| CUE (Cue for Playback) | 99 |
| CUER (Cue for Record) | 101 |
| EDIT (Edit) | 102 |
| FCLR (Clear Frames) | 102 |
| FF (Fast Forward) | 103 |
| FINS (Insert Frames) | 103 |
| FNEW (Insert New Frames) | 104 |
| FOVR (Overwrite Frames) | 104 |
| FRM (Remove Frames) | 105 |
| GET (Get Control) | 106 |
| GOTO (GO TO) | 107 |
| JOG (Jog) | 107 |
| LIMS (Set Edit Limits) | 108 |
| LOAD (Load Clip) | 109 |
| PAUS (Pause) | 110 |
| PLAY (Play Unit) | 110 |
| REC (Record) | 111 |
| REJU (Review) | 111 |

| | | |
|-----------|---|------------|
| | REW (Fast Reverse) | 112 |
| | RHRS (Rehearse) | 112 |
| | RSUM (Resume) | 113 |
| | SET (Set Control) | 114 |
| | SHTL (Shuttle) | 115 |
| | STOP (Stop Unit) | 115 |
| | UCLS (Unit Close) | 115 |
| | UERR (Unit Error) | 116 |
| | UFLS (Flush Unit) | 116 |
| | UGIN (Get Unit Info) | 117 |
| | UINF (Get Unit Info) | 117 |
| | UINT (Unit Interrupt) | 118 |
| | ULOC (Unit Location) | 118 |
| | UNLD (Unload Clip) | 119 |
| | UOPN (Unit Open) | 119 |
| | USTA (Unit Status) | 120 |
| | USYN (Set Unit Synchronization) | 121 |
| | UUWT (Unit Unit Wait) | 121 |
| | UWAT (Unit Wait) | 122 |
| A. | Using a MediaHub Archive System | 123 |
| | About MediaHub Clip Groups | 123 |
| | Bringing MediaHub Clips Into VCP-Recorder | 126 |
| | Finding MediaHub Clips | 127 |
| | Storing VCP-Recorder Clips in MediaHub | 128 |
| | Glossary | 129 |
| | Index | 133 |

List of Figures

| | | |
|-------------------|---|-----|
| Figure 1-1 | Functional Overview | 2 |
| Figure 1-2 | VCP-Recorder Media Control Panel | 3 |
| Figure 1-3 | VCP-Recorder Software Components | 5 |
| Figure 1-4 | One Logical Unit With One Control Connection | 9 |
| Figure 1-5 | One Logical Unit With Two Control Connections | 10 |
| Figure 1-6 | Two Logical Units With One Control Connection | 10 |
| Figure 1-7 | VCP-Recorder Hardware Components | 12 |
| Figure 2-1 | VCP-Recorder Media Control Panel | 19 |
| Figure 2-2 | The VCP-Recorder Media Control Panel With a Clip Loaded | 22 |
| Figure 2-3 | Loading a Clip Into the Logical VTR | 29 |
| Figure 2-4 | Create New Clip Window | 31 |
| Figure 2-5 | Deck Control Window | 34 |
| Figure 2-6 | Unit Status Monitor | 39 |
| Figure 3-1 | Clip Manager Window | 43 |
| Figure 3-2 | The Clip Information Window | 45 |
| Figure 3-3 | Rename Clip Window | 46 |
| Figure 3-4 | Delete Clip Window | 47 |
| Figure 3-5 | Set Protections Window | 48 |
| Figure 3-6 | Find Clip in Archive Window | 50 |
| Figure 3-7 | Get Clip from Archive Window: Getting a New Clip | 51 |
| Figure 3-8 | Get Clip from Archive Window: Updating an Existing Clip | 52 |
| Figure 3-9 | Put Clip to Archive Window | 53 |
| Figure A-1 | MediaHub Multiformat Clip Group | 125 |
| Figure A-2 | MediaHub Single-Format Clip Group | 126 |

List of Tables

| | | |
|------------------|-------------------------------|----|
| Table 2-1 | mcpanel Startup Options | 17 |
| Table 2-2 | Log Severity Levels | 18 |
| Table 3-1 | Log Severity Levels and Codes | 42 |
| Table 4-1 | MVCP Timecode Examples | 58 |
| Table 4-2 | MVCP Global Commands | 66 |
| Table 4-3 | MVCP Unit Commands | 97 |

About This Guide

VCP-Recorder is a digital media data playback and record engine for Silicon Graphics Origin servers and O2 workstations.

The *VCP-Recorder User's Guide* describes how to use VCP-Recorder to play and record digital media data and to store the data in, and retrieve it from, the MediaHub *archive system*. Also described in this document are the graphical user interface (GUI), which is used to manually control VCP-Recorder, and MVCP (Multi-Unit Video Computer Protocol), which is the control protocol used by VCP-Recorder.

What This Document Contains

The following material is covered in this document:

- Chapter 1, "Overview of VCP-Recorder," contains an overview of the product.
- Chapter 2, "Using the VCP-Recorder GUI," describes how to use the VCP-Recorder graphical user interface (GUI) to play and record clips and to determine status information.
- Chapter 3, "Using the Clip Manager," describes how to use the VCP-Recorder GUI to manage clips.
- Chapter 4, "MVCP Reference," describes the Silicon Graphics Multiple-Unit Video Computer Protocol.
- Appendix A, "Using a MediaHub Archive System," describes the information that is needed in order to locate *clips* in a MediaHub *archive system* or bring the clips from the archive system into VCP-Recorder.
- "Glossary" provides definitions of keywords used in this document.

Who Should Read This Document

This document is written for VCP-Recorder application developers and system integrators, and others who are interested in obtaining an overview of the product. It is assumed that the reader is already familiar with broadcast industry concepts.

Related Documentation

Refer to the following documents for related information:

- *VCP-Recorder Installation and Administration Guide*, for information about installing and maintaining a VCP-Recorder system
- *MediaHub User's Guide*, for information about using MediaHub, including the MediaHub Console application and the MediaHub command-line utilities
- *MediaHub Programmer's Guide*, for information about using the application programming interface for the MediaHub archive system

Refer to the following documents for supplementary information:

- *IRIX Admin: Software Installation and Licensing* (part number 007-1364-*nnn*) for information about installing software that runs under IRIX, the Silicon Graphics implementation of the UNIX operating system
- *IRIX Admin: System Configuration and Operation* (part number 007-2859-*nnn*) for information about IRIX system administration tasks
- *IRIX Admin: Disks and Filesystems* (part number 007-2825-*nnn*) for information about general filesystem concepts and system administration procedures for SCSI disks, XFS and EFS filesystems, logical volumes, and guaranteed rate I/O

Conventions Used in This Document

The following type and symbol conventions are used in this document:

Italics Used for filenames, pathnames, directory names, emphasis, document titles, variable names, glossary terms, and command-line programs

Bold Used for keywords

Fixed-width Used for code examples and command syntax

Bold fixed-width
 Used for user input, including nonprinting keyboard keys

Square brackets ([]) Surround syntax statement arguments that are optional

Square bullets (■) Indicate substeps within a multistep process

Ellipsis (...) Indicate that the preceding is repeated

Right angle brackets (>) Indicate a path through menus to a menu option. For example, “File > Open” means “Under the File menu, choose the Open option.” Right angle brackets also indicate the play button in the graphical user interface.

Overview of VCP-Recorder

This chapter contains an overview of VCP-Recorder, which is a software product used by application developers and systems integrators to enable a Silicon Graphics Origin server or O2 workstation to be used as a video server. VCP-Recorder provides real-time, frame-accurate recording and playback of broadcast-quality digital media data.

Note: The term *Origin* in this document refers to Origin200, Origin2000, and Onyx2 servers. Where there is a distinction, the pertinent product name is used.

The following topics are discussed in this chapter:

- “Functional Overview” on page 1
- “Software Overview” on page 4
- “Hardware Overview” on page 11

Functional Overview

Digital media data is brought into VCP-Recorder by recording it from a live feed or a videotape deck, retrieving it from an *archive system*, or copying it from a file. The data can then be played out to a broadcast system, a video port, or a videotape deck; sent to an MPEG-2 decoder for playout; or transferred to an archive system for storage and distribution.

VCP-Recorder can be automatically controlled by an application program or through the use of a broadcast system *automation controller*. VCP-Recorder can also be manually controlled by using the VCP-Recorder graphical user interface or a telnet connection using the Multiple-Unit Video Computer Protocol (*MVCP*).

This functionality is shown in Figure 1-1.

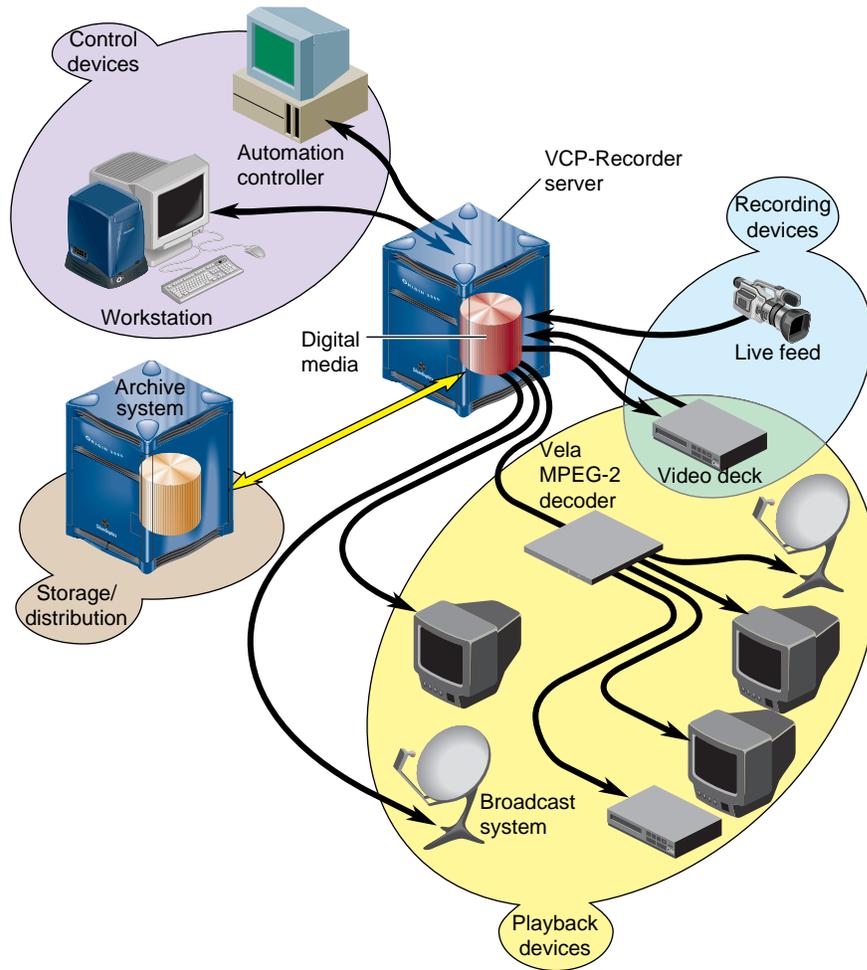


Figure 1-1 Functional Overview

The graphical user interface (GUI), which is used to control VCP-Recorder, was designed as both a demonstration of the VCP-Recorder capability and a beginning point from which broader graphical applications can be developed. The GUI consists of screens that are used to record and play digital data, determine status information, and manage digital media data stored in VCP-Recorder.

The GUI screen that is used to record and play digital media data is the VCP-Recorder Media Control Panel, which is shown in Figure 1-2. The control panel is similar in function to a standard videotape player/recorder. For example, there are buttons in the control panel to cue the video, play it, stop the playback, and so on.

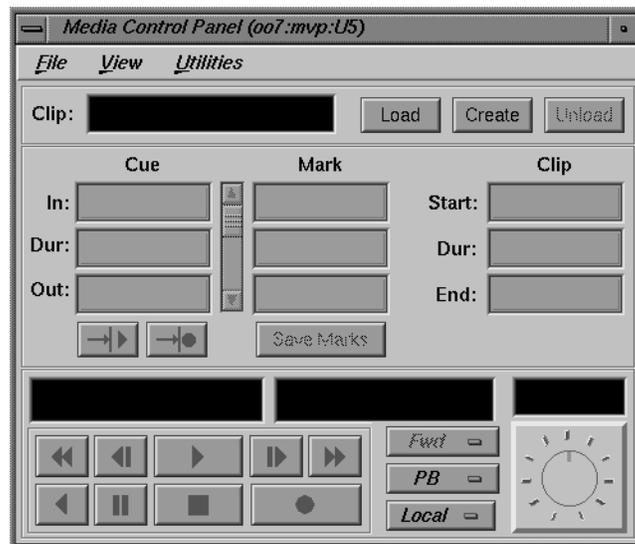


Figure 1-2 VCP-Recorder Media Control Panel

The graphical user interface is described in Chapter 2, "Using the VCP-Recorder GUI." The MVCP protocol, which was used to implement the VCP-Recorder GUI, is described in Chapter 4, "MVCP Reference."

Software Overview

The VCP-Recorder software provides scalability and maximum flexibility, while enabling real-time, frame-accurate control of digital media. The software includes the following:

- Core software, which provides the basic VCP-Recorder functionality for playback and recording of digital data.
- Control interface modules, which provide device-dependent code. For example, there is a control interface module that contains the code that is specific to a Louth *automation controller*.
- Media device interface modules, which contain format-dependent code that provides access to the *ports* over which media is played and recorded. For example, there is a media device interface module that contains the code that supports the Vela Research SCSI-attached MPEG-2 decoder.
- Format interface modules, which provide handlers for accessing specific digital media storage formats. For example, there are format interface modules for the VCP-Recorder variable-frame format and the MPEG-2 stream-based format.

Figure 1-3 shows the primary software components in VCP-Recorder.

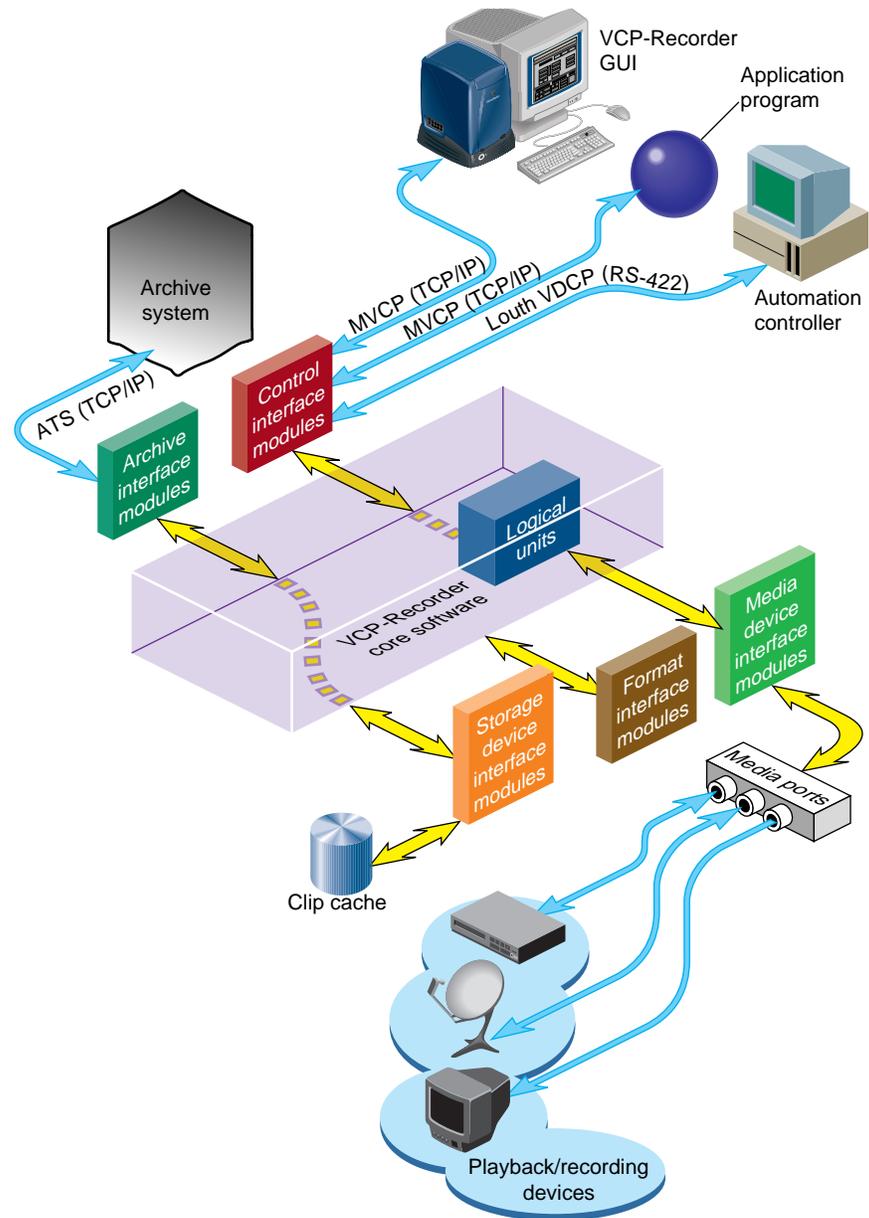


Figure 1-3 VCP-Recorder Software Components

The remainder of this section discusses the software components shown in Figure 1-3.

Clip Cache

Digital media data that VCP-Recorder processes for playout and recording is stored in one or more *clip caches*. Each unit of data that is stored in a clip cache (for example, a movie) is called a *clip*.

Clips can be transferred into VCP-Recorder from an attached *archive system*, such as MediaHub, or they can be recorded by VCP-Recorder. Clips that are recorded by VCP-Recorder can then be transferred from the clip cache into an attached archive system for storage.

Core Software

The core software provides the basic VCP-Recorder functionality for playback and recording of digital data. It utilizes the IRIX operating system as well as portions of the Silicon Graphics Digital Media Libraries.

The core software provides the following basic functionality:

- archive management, which oversees the transferring of *assets* into and out of an attached *archive system*
- clip cache management, which maintains persistent information about the media that is either stored in the *clip cache* or is in the process of being transferred into or out of it
- controller management, which links one or more external control protocol modules (for example, Louth) to the internal VCP-Recorder processing logic
- configuration management, which automatically configures the VCP-Recorder software according to the hardware capabilities of the system on which it runs

VCP-Recorder provides a core library that supports external interface modules, which are dynamic shared objects (DSOs) that contain the code that is specific to a given external entity. For example, the MediaHub archive interface module has the code that is specific to the MediaHub archive system. When VCP-Recorder is started, the VCP-Recorder software loads and initializes all external interface modules that it locates so that the modules can be used.

Archive Interface Module

Archive interface modules contain the code that is specific to a given external *archive system*. These modules provide the support that is needed to locate a given *clip* in the archive system and bring it into VCP-Recorder, and to store a clip from VCP-Recorder in the archive system.

VCP-Recorder has an archive interface module for MediaHub, which enables VCP-Recorder to use the MediaHub Asset Transfer Service (ATS) to retrieve *clips* from, and store them in, a MediaHub site. VCP-Recorder transfers the media to and from MediaHub using the ATS protocol and TCP sockets in a manner that is similar to the File Transfer Protocol (FTP).

Control Interface Module

Control interface modules allow various *automation controllers* and digital media applications to control the use of VCP-Recorder. These modules translate to and from external control protocols.

The following control interface modules are provided:

- The Louth Video Disk Communications Protocol, which is defined by Louth Automation. This control protocol provides control of VCP-Recorder over RS-232, RS-422, or TCP/IP. The VCP-Recorder's Louth interface module supports back-to-back play and record (subject to restrictions imposed by the video I/O port capabilities) and archive management.
- Multiple-Unit Video Computer Protocol (MVCP), which is defined by Silicon Graphics. This control protocol provides full-featured control of VCP-Recorder through TCP/IP. This control interface module supports archive management, multiple-unit control, and event monitoring, and provides access to advanced features of Silicon Graphics devices.

Storage Device Interface Module

Storage device interface modules provide access to the storage systems on which the *clip cache* resides. Currently, there is a storage device interface module for the IRIX XFS filesystem.

Format Interface Module

Format interface modules provide handlers for accessing specific digital media storage formats. VCP-Recorder currently provides format interface modules that support the following:

- VCP-Recorder variable-frame format (uncompressed, *Rice* compression, and motion JPEG)
- MPEG-2 stream-based format (transport and program streams)

Media Device Interface Module

Media device interface modules provide access to the *ports* over which the media is played and recorded (that is, the media ports). Each type of I/O port typically has its own media device interface module.

VCP-Recorder has media device interface modules for the following:

- Silicon Graphics Video Library. This media device interface module supports the Digital Video Option (*DIVO*) on Origin servers and the video port on O2 workstations.
 - *DIVO* enables broadcast-quality video and embedded audio, and is used for the recording and playback of uncompressed and *Rice* compression formats.
 - The O2 video port is used for playback and recording of the motion JPEG format, and is suitable for a test or development environment.

Note: The O2 video port is usually referred to as *mvp* (multiport video processor) or *O2Video*.
- Vela Research 4-port analog decoders. This media device interface module enables the playback of MPEG-2 format data on Origin servers and O2 workstations.
- VLAN transmitters. This media device interface module enables frame-accurate capture from, and lay-down to, videotape decks.

Logical Playback and Record Units

Logical *units* enable media *ports* to play and record *clips*. Each VCP-Recorder unit can be thought of as a logical videotape recorder (VTR) transport that is capable of loading, cueing, playing, and recording clips using a specific media port.

Logical units are created automatically by VCP-Recorder when the VCP-Recorder GUI or an *automation controller* is used. When the *MVCP* protocol is used, a command requests that a unit be created or that a unit created by another control connection be used.

There is normally a one-to-one relationship between a control connection to VCP-Recorder and a VCP-Recorder logical unit, and between a logical unit and a media port. This is shown in Figure 1-4.

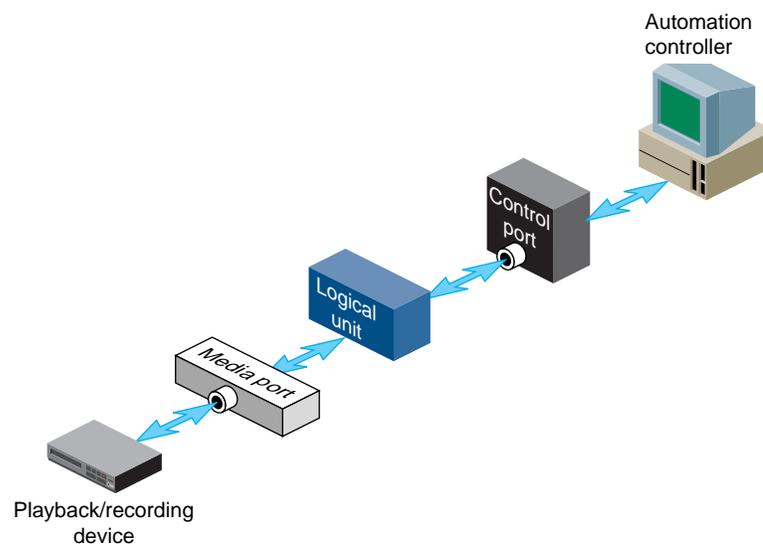


Figure 1-4 One Logical Unit With One Control Connection

A single unit can also be controlled by multiple control ports. For example, two tightly integrated applications might control a single unit, where each application would have its own control port. This is shown in Figure 1-5.

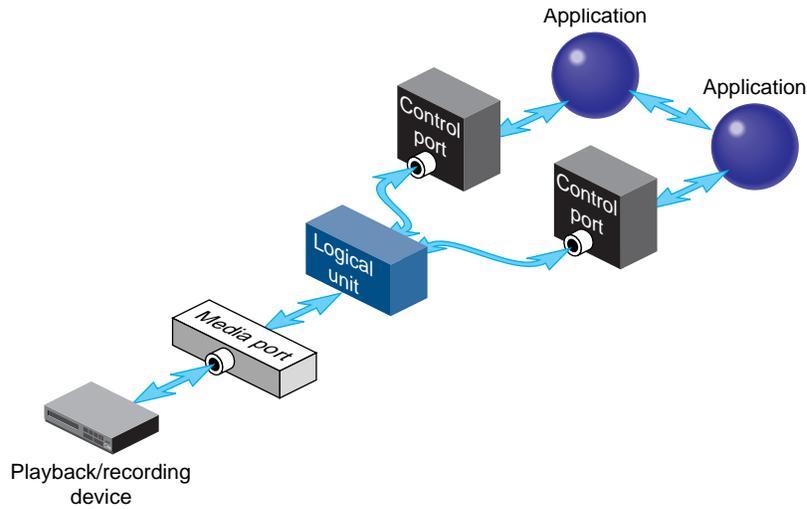


Figure 1-5 One Logical Unit With Two Control Connections

In addition, a media port can be controlled by multiple logical units. For example, an application with one control connection and two units could be cueing one clip while playing out another, enabling back-to-back playout of clips. This example is shown in Figure 1-6.

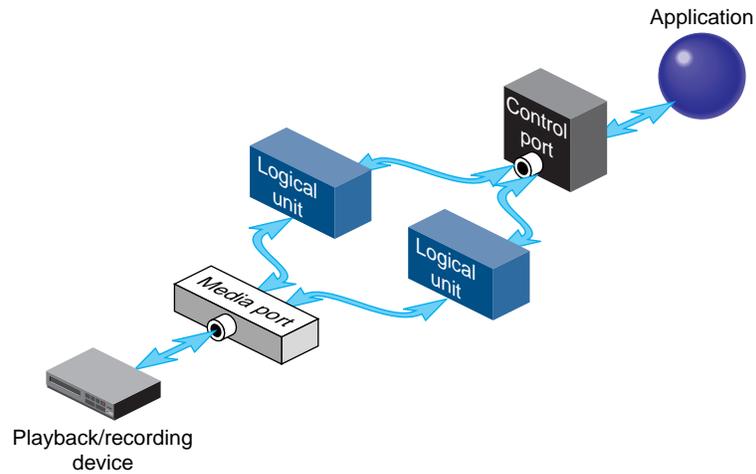


Figure 1-6 Two Logical Units With One Control Connection

If a media port supports multiple logical units, the sharing is subject to the device-sharing characteristics of that port.

Caution: Extreme care must be exercised when the control of a unit is shared between two controller connections. Interfering with a unit owned by a Louth *automation controller* leads to unpredictable behavior.

Hardware Overview

A VCP-Recorder hardware configuration consists of a server; disk storage for the *clip cache*; and various Silicon Graphics and select third-party external devices for recording and playing out *clips* and controlling VCP-Recorder. A typical hardware configuration is shown in Figure 1-7.

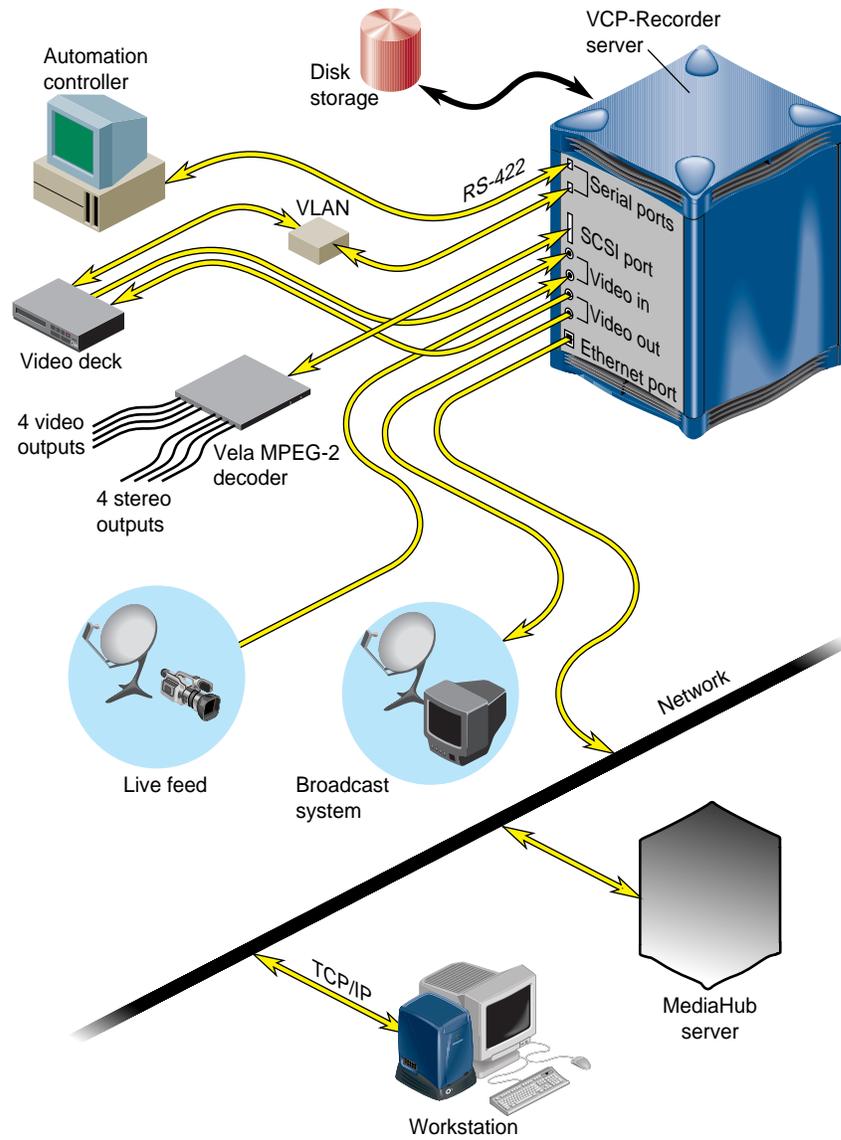


Figure 1-7 VCP-Recorder Hardware Components

VCP-Recorder Server

The VCP-Recorder server can be an Origin200 or Origin2000 Scalable Symmetric Multi-Processing (S2MP) server, or an O2 workstation.

Origin Servers

The Origin servers provide massive processing, storage, and throughput capabilities to satisfy even the largest production requirements. They are built from a scalable node architecture, enabling small configurations that can be incrementally upgraded to the largest possible configurations. Each Origin server can be configured as a single module or as multiple modules with a single system image.

The Origin servers provide:

- playback and recording of uncompressed and *Rice* compression video data with the Digital Video Option (*DIVO*) card
- playback of MPEG-2 (system and transport streams) data with the Vela Research SCSI-attached decoder

O2 Workstations

The O2 workstation provides a configuration that is an excellent evaluation, development, or test system, but is not recommended for use in a production environment.

The O2 workstation provides

- playback and recording of M-JPEG video data
- playback of MPEG-2 (system and transport streams) data with the Vela Research SCSI-attached decoder

Disk Storage

The VCP-Recorder disk storage holds the XFS real-time filesystems that contain the movies, trailers, commercials, and other digital media data stored in the *clip cache*. The descriptive information about VCP-Recorder *clips*, for example, clip names, duration, *edit points*, and so on, are also stored on these filesystems.

The VCP-Recorder storage system supports the use of scalable storage to enable the total disk space to range from only a few gigabytes to hundreds of terabytes or more. The type of disk storage that is used depends upon several factors, including the number and size of stored clips, the use of RAID, and the required availability (uptime) of the system.

Three different types of disk storage are available:

- Normal XFS filesystems residing on a single disk drive. This type of disk storage does not provide redundancy.
- Standard disk storage, sometimes called RAID-0, in which several disk drives are striped into XLV logical volumes. This type of disk storage does not provide redundancy.
- RAID storage, such as RAID-3 and RAID-5, which provides high-availability, redundant digital storage.

External Devices

VCP-Recorder supports the use of the following external devices:

- Vela Research 4-port MPEG-2 decoders, which are used for the playback of MPEG-2 format data. Vela decoders are connected to the VCP-Recorder server through a SCSI connection.
- VLAN transmitters, which are used to control a videotape deck for frame-accurate capture and lay-down. A VLAN transmitter is connected to a VCP-Recorder server through an RS-422 serial port connection.
- Broadcast system *automation controllers* that use the Louth Video Disk Communications Protocol, which is defined by Louth Automation. Automation controllers are connected to the VCP-Recorder server through an RS-422 serial connection or a TCP/IP ethernet connection.

Using the VCP-Recorder GUI

The VCP-Recorder graphical user interface (GUI) was designed as both a demonstration of the VCP-Recorder capability and a beginning point from which broader graphical applications can be developed. The GUI consists of the following:

- Media Control Panel, which enables *clips* to be played and recorded
- Deck Control Panel, which enables a video deck to be controlled
- Unit Status Monitor, which displays the status of VCP-Recorder ports
- Clip Manager, which is used to manage clips, including getting them from, and writing them to, an *archive system*

The following topics are discussed in this chapter:

- “Playing and Recording Clips” on page 16
- “Monitoring the Status of VCP-Recorder” on page 39

Chapter 3, “Using the Clip Manager,” describes how to use the GUI to manage clips. Chapter 4, “MVCP Reference,” describes the *MVCP* protocol, which was used to implement the VCP-Recorder GUI.

Playing and Recording Clips

This section describes how to use the VCP-Recorder Media Control Panel and how to control a video deck. The following topics are discussed:

- “Starting the VCP-Recorder Media Control Panel” on page 16
- “About the Media Control Panel” on page 22
- “Playing or Recording an Existing Clip” on page 29
- “Creating a New Clip” on page 31
- “Changing Cue Points and Edit Points” on page 32
- “Controlling a Video Deck” on page 33

See the *VCP-Recorder Installation and Administration Guide* for information about how to copy digital media data from a file into the VCP-Recorder *clip cache*.

Starting the VCP-Recorder Media Control Panel

To establish a control connection to VCP-Recorder and start the Media Control Panel, enter the following, either from the workstation on the VCP-Recorder server or from a workstation on which the VCP-Recorder tools software¹ has been installed:

```
% /usr/vtr/bin/mcpanel [-h hostname] [-p videoPort|unit] [-D deckCtlPort]
[-c clipname] [-r] [-C "inpoint outpoint"] [-P] [-v loglevel]
```

¹To run the Media Control Panel from a remote workstation, the `vcp_recorder_eoe.sw32.tools` subsystem must be installed on a workstation that has IRIX 6.2 or later. See the *VCP-Recorder Installation and Administration Guide* for more information.

Table 2-1 describes each of the options available when starting the VCP-Recorder Media Control Panel.

Table 2-1 mcp panel Startup Options

| Option | Description |
|--------|---|
| -c | Identifies the name of a clip to be loaded when the Media Control Panel starts. If this option is not specified, no clip is initially loaded. |
| -D | Specifies the video deck control port (<i>deckCtlPort</i>) to be used for VLAN communication to an external video storage device, such as a digital videotape recorder (VTR). The deck is controlled by the Deck Control Panel. If this option is not specified, deck control is not available. |
| -h | Identifies the host on which VCP-Recorder runs. This enables the Media Control Panel to be run from a remote workstation. If this option is not specified, the local host is assumed. |
| -p | Identifies the VCP-Recorder host video <i>port</i> or <i>unit</i> to which the control connection is made. <i>videoPort</i> is the VCP-Recorder host video port to which the control connection is made. If you specify a port, VCP-Recorder creates a new logical unit that is used by this control connection. <i>unit</i> is the VCP-Recorder logical unit to which the connection is made. If you specify a unit, it must have already been created by another control connection. The control connection being made shares the unit with the control connection that created the unit. (Units are named with a capital “U” followed by a number, for example, U4.) If this option is not specified, the first video port on the VCP-Recorder is used. |
| -r | Specifies that if an mcp panel already exists for the video port, the existing mcp panel should be raised on the desktop instead of creating a new one. |
| -C | Specifies that the loaded clip should be cued with the specified in- and out-points. If “*” is specified for either <i>inpoint</i> or <i>outpoint</i> , the default edit in-point or out-point is used. |
| -P | Specifies that the clip whose name is <i>clipname</i> should start playing when the Media Control Panel starts. |
| -v | Sets the logging verbosity level to <i>loglevel</i> . The default is 0, meaning all log messages up to and including Info priority are written to STDOUT. (The mcp panel program writes its log messages to STDOUT.) The values for <i>loglevel</i> are defined in Table 2-2. |

Table 2-2 shows the log severity levels and codes, which are listed in decreasing order of severity

Table 2-2 Log Severity Levels

| Priority | Log Level | Description |
|----------------|-----------|--|
| Emergency | -6 | Panic condition |
| Alert | -5 | Condition that should be corrected immediately, such as a corrupted system file |
| Critical | -4 | Critical condition that has system-wide impact, such as a hard device error; immediate action required |
| Error | -3 | Problem that needs correcting but does not require immediate action |
| Warning | -2 | Possible problem but could be a transient problem that corrects itself |
| Notice | -1 | Condition that might require attention, but is not an error condition |
| Info | 0 | Informational message |
| Debug <i>n</i> | <i>n</i> | Informational message that normally is of use to engineers for debugging; may be Debug1, Debug2, or Debug3, with Debug3 producing the most debugging information |

The VCP-Recorder Media Control Panel, shown in Figure 2-1, is displayed in a separate window. This control panel represents a unit, or logical *VTR*, that is used to play and record *clips*. The buttons in the Media Control Panel are similar in function to those of a standard videotape player/recorder. For example, there are buttons to load a clip, play it, and pause.

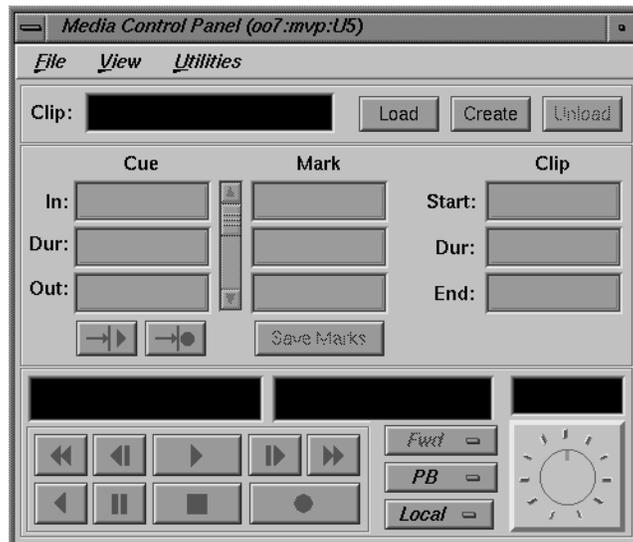


Figure 2-1 VCP-Recorder Media Control Panel

See “About the Media Control Panel” on page 22 for a detailed description of the Media Control Panel.

To determine which video and deck control ports are available on a VCP-Recorder server, establish a telnet connection to the VCP-Recorder server and then use the MVCP *PLS* (List Ports) command. (See “Establishing an Interactive MVCP Connection” on page 56 for details.) The following example shows the type of information returned by *PLS* for an O2 server. This example identifies “mvp” as the video port (VID) on the server:

```
PLS
201 OK
mvp BOTH "SGI O2 (MACE) Video Processor" VID
```

The following example shows the *PLS* command output for an Origin server. This output identifies “vlan_1” as the deck control port (DECK) and “DIVO_n,” where *n* is 0-7, as the video ports (VID):

```
PLS
201 OK
vlan_1 BOTH "VLAN Deck Control" DECK
DIVO_0 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_1 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_2 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_3 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_4 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_5 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_6 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_7 BOTH "SGI XT-DIVO Digital Video Option" VID
```

See “PLS (List Ports)” on page 91 for more information.

You can also determine the ports by using the `/usr/vtr/bin/vtrstat` command:

```
% /usr/vtr/bin/vtrstat -ports
# Port      Type      Description
-----
0  mvp       Video     SGI O2Video (Multiport Video Processor)
```

See the `vtrstat(1)` man page for more information.

To determine which units are in use on a VCP-Recorder server, establish a telnet connection to the VCP-Recorder server and then use the *MVCP ULS* (List Units) command. The following example shows the information returned by the *ULS* command:

```
ULS
201 OK
U1 mvcp/originserver mvp BOTH * DONE IDLE * 0 *
U2 mvcp/originserver mvp BOTH * DONE IDLE * 0 *
```

This example indicates that there are two units (U1 and U2) on the server, both using the `mvcp` port on the host named `originserver`. The units were opened for input and output (BOTH) and they are currently idle.

See “ULS (List Units)” on page 96 for more information.

You can also determine the units by using the `/usr/vtr/bin/vtrstat` command:

```
% /usr/vtr/bin/vtrstat -units
Unit  Owner      Port   Clip      Function Location
-----
U3    mvcp/oo7     mvp    *          IDLE     *
```

See the `vtrstat(1)` man page for more information.

The following are examples of starting the Media Control Panel:

- When the Media Control Panel is started by entering the following command, a control connection is made to the `DIVO_1` video *port* on the “`origin_server`” host using a newly added unit. Messages with a severity level of Info and above are written on STDOUT:


```
% /usr/vtr/bin/mcpanel -h origin_server -p DIVO_1
```
- When the Media Control Panel is started by entering the following command, a control connection is made to the `U9` unit, which must already exist. Messages with a severity level of Debug2 and above are written on STDOUT:


```
% /usr/vtr/bin/mcpanel -v 2 -h origin_server -p U9
```
- When the Media Control Panel is started by entering the following command, a control connection is made to the `mvp` video port on the “`o2server`” host. Messages with a severity level of Info and above are written on STDOUT:


```
% /usr/vtr/bin/mcpanel -h o2server -p mvp
```

About the Media Control Panel

Figure 2-2 shows the appearance of the Media Control Panel after a *clip* has been loaded. The header of the control panel identifies the host, control *port*, and *unit* to which the control panel is connected.

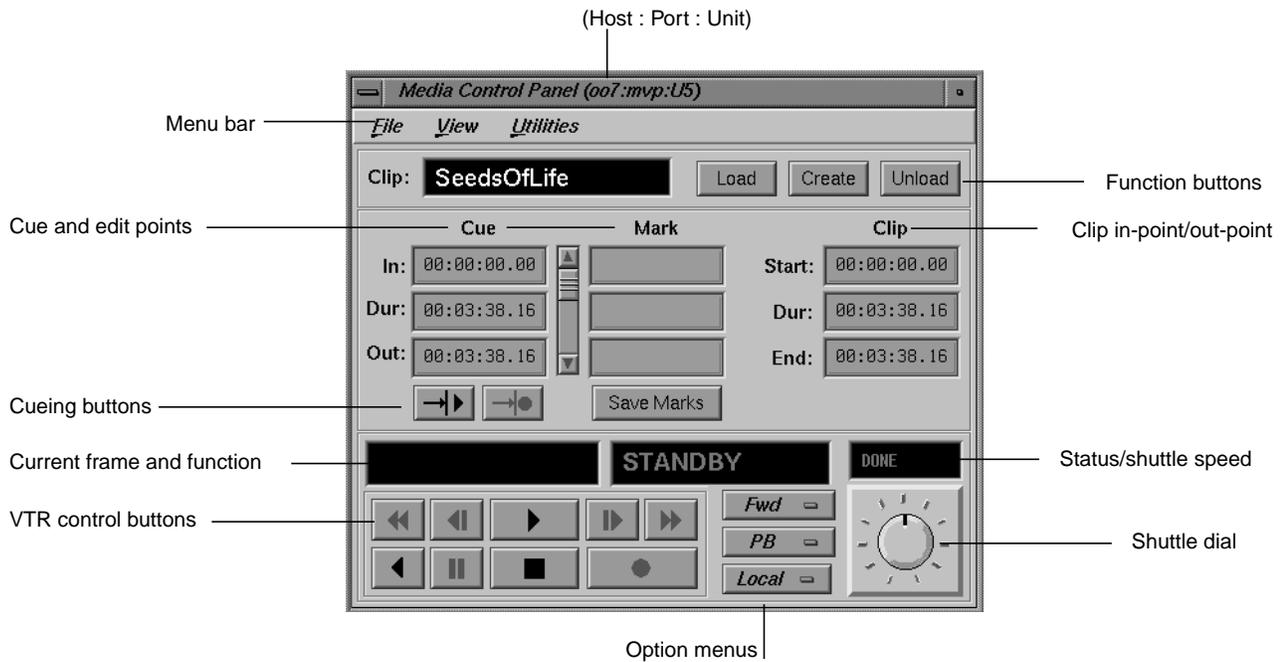


Figure 2-2 The VCP-Recorder Media Control Panel With a Clip Loaded

The menu bar gives you access to the following:

- The File pulldown menu, which lets you load or unload an existing *clip*, create a new clip, or close the Media Control Panel. Most of the functions that are available in this pulldown menu are available through buttons in the Media Control Panel.
- The View pulldown menu, which lets you access the Deck Control window. The Deck Control window, which is shown in Figure 2-5, is used to control a video deck that is attached to the VCP-Recorder host. See “About the Deck Control Window,” on page 34, for more information.

- The Utilities pulldown menu, which lets you access the following:
 - the IRIX audio panel
 - the IRIX video panel
 - the VCP-Recorder Unit Status Monitor (described in “Monitoring the Status of VCP-Recorder” on page 39)
 - the VCP-Recorder Clip Manager (described in Chapter 3, “Using the Clip Manager”)

The following describes each of the displays and buttons in the Media Control Panel:

- The Clip field contains the name of the *clip*, if one is loaded. The display is blank if a clip is not loaded.
- The *Load*, *Create*, and *Unload* function buttons let you load an existing clip, create a new one and record into it, and unload a clip, respectively.
- The *cue points* are used to move around within a clip and to control the portion of the clip that is played. An in-point (In), the duration (Dur), and an out-point (Out) are specified using the following format:

hh:mm:ss:ff

where *hh* is the hours, *mm* is the minutes, *ss* is the seconds, and *ff* is the frame number. In drop-frame mode, the final colon is replaced by a period:

hh:mm:ss.ff

For example, if a clip with a cue in-point of 00:00:30.00 is cued for playing, it is cued at thirty seconds.

See “Changing Cue Points and Edit Points” on page 32 for information about how to change the cue points.

- The *edit points* (under “Mark” in the Media Control Panel) are persistent values stored with a clip. They are used to initialize the *cue points* when the clip is loaded. An in-point (In), the duration (Dur), and an out-point (Out) are specified using the following format:

hh:mm:ss:ff

where *hh* is the hours, *mm* is the minutes, *ss* is the seconds, and *ff* is the frame number. In drop-frame mode, the final colon is replaced by a period:

hh:mm:ss.ff

Note: Edit points may also be called “edit marks.”

If a clip has edit points associated with it, those values are used to initialize the clip’s cue points when the clip is loaded.

See “Changing Cue Points and Edit Points” on page 32 for information about how to change the edit points.

- To the right of the cue and edit points are the start, the duration (Dur), and end of the clip. Each is specified in the following format:

hh:mm:ss:ff

where *hh* is the hours, *mm* is the minutes, *ss* is the seconds, and *ff* is the frame number. In drop-frame mode, the final colon is replaced by a period, as in:

hh:mm:ss.ff

If a clip does not have edit points associated with it, these start, duration, and end values are used to initialize the cue points when the clip is loaded.

Note: You cannot change the values of the start, end, or duration of the clip, itself. However, you can select any of the values and copy it to the cue points or edit points.

- The cue buttons cue the clip for playout (--> | >) or recording (--> | •). The location at which the clip is cued depends on how the clip is cued and the play direction, as discussed in the descriptions of the *VTR* control buttons and the Media Control Panel’s option menus. (The play direction is determined by the setting of the topmost option menu.)

Note: A clip must be cued before it can be played or recorded. This can be accomplished explicitly by clicking a cue button or implicitly by clicking the play or record button without first clicking a cue button. If you click a cue button and then click play or record, the clip starts playing or recording immediately. If you click play or record without first clicking a cue button, the Media Control Panel first cues the clip and then starts the requested function. In the latter case, there is a brief delay before the requested play or record function begins.

- The current frame display is initialized to the cue in-point or cue out-point each time the clip is cued. The actual *cue point* depends on how the clip is cued and the clip's play direction, as discussed in the descriptions of the *VTR* control buttons and the play direction. (The play direction is determined by the setting of the topmost option menu.) As the clip is played or recorded, the display changes to indicate the current frame number.
- The function display shows the current function. When a clip is first loaded, the word "STANDBY" is displayed.
- When a clip is being played or recorded, the status/shuttle speed display shows the speed. If the clip is not being played or recorded, the status is displayed. The status may be one of the following:
 - BUSY, when the function is waiting to execute or waiting for another *unit* to finish (for example, a Louth *automation controller* that is playing a clip). The word BUSY blinks on and off.
 - RUN, when the function is in progress.
 - DONE, when the function completed without an error. The word DONE and the function display are grayed-out.
 - ERROR, when an error has occurred. The word ERROR blinks on and off, and appears in red.

- The *VTR* control buttons control the playout and recording of clips. The following describes these buttons, which correspond to standard VTR buttons:
 - Fast-reverse (<<) plays the clip in reverse at a fast speed.
 - Jog backward (<|) jogs the clip backward by one frame. Each time you click the jog backward button, the clip jogs back one frame.

If the clip is playing when you click this button, the clip jogs backward one frame and then pauses. You have to click either the play button or the pause button to resume play.
 - Forward play (>) plays the clip in the forward direction. If the clip is not cued, it is cued before it begins playout.
 - Jog forward (|>) jogs the clip forward by one frame. Each time you click the jog forward button, the clip advances one frame.

If the clip is playing when you click this button, the clip jogs forward one frame and then pauses. You have to click either the play button or the pause button to resume play.
 - Fast-forward (>>) plays the clip forward at a fast speed.
 - Reverse play (<) plays the clip in reverse. If the clip is not cued, it is cued before it begins playout.
 - Pause (| |) temporarily stops the clip from playing or recording. You have to click the play, record, or pause button to resume.
 - Stop (■) stops the clip and de-cues it. After the clip has been stopped, you must re-cue it to play it again.
 - Record (•) begins recording. If the clip is not cued, it is cued for recording before it begins.

- The top option menu lets you specify the play direction, which determines the direction in which the clip is played and whether it plays once or plays until it is stopped. The following options are available through this menu:
 - *Fwd*, for forward play (default)
 - *F Lp*, for forward loop play
 - *Bwd*, for backward play
 - *B Lp*, for backward loop play
 - *F/Bwd*, for alternating forward and backward play
 - *F/B Lp*, for alternating forward and backward loop play
 - *B/Fwd*, for alternating backward and forward play
 - *B/F Lp*, for alternating backward and forward loop play

If the direction is one of the forward directions (*Fwd*, *F Lp*, *F/Bwd*, or *F/B Lp*), the clip is cued at its in-point. If the direction is one of the backward directions (*Bwd*, *B Lp*, *B/Fwd*, or *B/F Lp*), the clip is cued at its out-point.

The forward direction means that the clip plays from its in-point to its out-point. The backward direction means that the clip plays from its out-point to its in-point. For alternating directions, the clip plays once in each direction. In loop mode, the clip continues to play in the given direction until you click the stop button. If you do not choose loop mode, the clip plays once in the indicated direction and then stops.

Note: When you change the option in this menu, the change takes effect the next time you cue the clip. For example, assume that the *Fwd* option is in effect when you start playing a clip. If you choose the *F Lp* option after the playing starts, the play stops when the out-point is reached. The next time you play the clip, the clip plays in forward loop mode.

- The middle option menu lets you choose the following options:
 - *PB*, the clip's audio and video are output when a clip is playing; nothing is output at other times (default).
 - *PB/EE*, the clip's audio and video are output when a clip is playing; the input signal is output at other times.

Note: The EE (end-to-end) option emulates a video deck feature and is relevant only when you have an active input source.
 - *PB/Im*, the clip's audio and video are output when a clip is playing; SMPTE 75% colors bars and 1kHz tone are output at other times.
 - *PB/B*, the clip's audio and video are output when a clip is playing; a black screen is output at other times.
 - *EE*, the output always displays the input signal instead of the signal from VCP-Recorder, even when a clip is playing. (When a clip is playing, the output displays the input signal.)
 - *Image*, the output always displays a color bar image.
 - *Black*, the output always displays a black screen.
 - *Hold*, the output always displays the last image.
- The Local/Rem option menu lets you put the *unit* in remote mode. The following options are available through this menu:
 - *Local*, which puts the unit in local mode and enables the *VTR* control buttons (local).
 - *Rem*, which puts the unit in remote mode and disables the *VTR* control buttons. This mode prevents you from accidentally operating a Media Control Panel while the unit is being controlled remotely, for example, by an *automation controller*.

- The shuttle dial lets you control the speed of a clip that is playing. To use the shuttle dial, begin playing the clip and then use the mouse to point to the black notch on the dial. Press the left mouse button and keep it pressed while you turn the dial clockwise to increase the speed or counterclockwise to decrease it.

Note: The straight up position of the dial is zero, or pause.

In general, buttons that are enabled and can be used appear darker than those that are disabled. For example, when you load an existing clip, the cue for playout button (--> | >) is darker than the cue for recording button (--> | •). This indicates that the cue for playout button is enabled and the cue for recording button is not.

Playing or Recording an Existing Clip

To play or record an existing *clip*, follow these steps:

1. If the clip is not loaded, load the clip:
 - Click the *Load* button in the Media Control Panel. A window that lists the clips managed by VCP-Recorder appears, as shown in Figure 2-3.

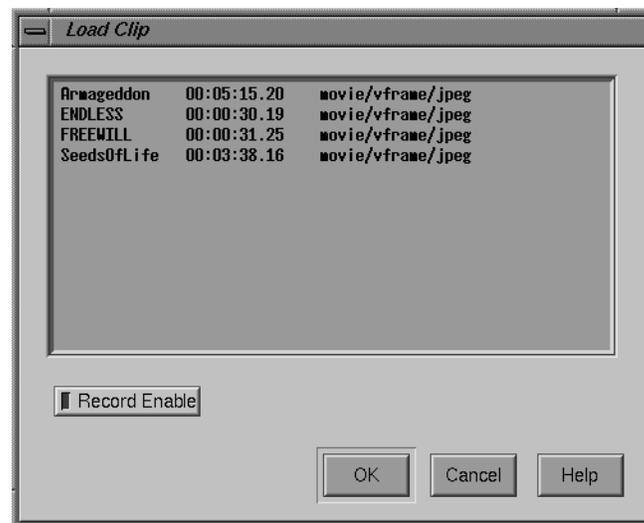


Figure 2-3 Loading a Clip Into the Logical VTR

The Load Clip window lists the name, duration, and format of each clip that is stored in `/usr/vtr/clips`, which is the VCP-Recorder *clip cache*. For example, the SeedsOfLife clip has a duration of 00:03:38.16 and its format is `movie/vframe/jpeg`. (If the duration contains an “*” it means that no material has been recorded in the clip.)

Note: If a clip is stored in a directory within `/usr/vtr/clips`, that directory name precedes the clip name. For example, if the clip name appears as `ADS/COMM1`, the clip is stored in `/usr/vtr/clips/ADS/COMM1`.

- Select the clip that you want to load. (You select the clip by pointing to it with the mouse cursor and pressing the left mouse button.)
 - If you want to record over this clip, click *Record Enable*. (When recording is enabled, the yellow LED is lit in the *Record Enable* button.) Click *Record Enable* a second time if you want to disable this option.
 - Click the *OK* button. The window closes and the selected clip appears in the Media Control Panel. The cue for playout button (`--> | >`) is enabled. If the clip can be recorded over, the cue for recording button (`--> | •`) is also enabled.
2. Change the *cue points*, as needed. (See “Changing Cue Points and Edit Points” on page 32 for information.)
 3. Click the cue for playout button (`--> | >`) or the cue for recording button (`--> | •`).

Note: If you cue the clip first, the clip starts playing or recording as soon as you click the play or record button. If you do not cue the clip beforehand, the Media Control Panel automatically cues the clip and then starts playing or recording it. In the latter case, there is a delay before the requested function begins.

4. Click the play button (`>`) to start the playout or the record button (`•`) to start recording.
5. To stop playing or recording, click the stop button (`■`).

Note: If you want to replay or re-record the clip after it stops, the clip must be re-cued. If you click the play or record button without first clicking the cue button, the Media Control Panel automatically cues the clip before it starts playing or recording.

Creating a New Clip

To create a new *clip* and record its content, follow these steps:

1. Click the *Create* button in the Media Control Panel. The Create Clip window, shown in Figure 2-4, appears.

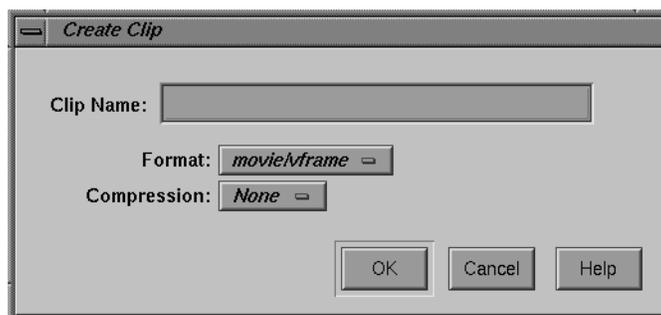


Figure 2-4 Create New Clip Window

2. Enter the name of the clip in the Clip Name field.
3. Choose the format of the movie from the *Format* option menu.

Note: You can create clips through the VCP-Recorder graphical user interface using only the “movie/vframe” option, which is the VCP-Recorder variable-frame movie format.
4. Choose the compression type of the clip from the *Compression* option menu. You can choose from the following compression types:
 - Rice (*Rice* compression; only on Origin servers with the Digital Video Option)
 - None (uncompressed; only on Origin servers with the Digital Video Option)
 - JPEG (M-JPEG; only on O2 workstations)
5. Click the *OK* button. The window closes and the clip appears in the Media Control Panel. Both the cue for recording button (--> | •) and the cue for playback button (--> | >) are enabled.
6. Click the cue for recording button (--> | •).
7. To start recording, click the record button (•).
8. To stop recording, click the stop button (■).

Changing Cue Points and Edit Points

You can change the *cue points* and *edit points* of a clip in the Media Control Panel in three ways:

- Select and type a new value. For example, if you want to change a value from 00:00:00.01 to 00:00:00.05, select the “1” and type “5.”
- Click inside a field and use the scroll bar. The Media Control Panel automatically adjusts the other entries, accordingly, when you use the scroll bar. For example, to increase the cue in-point, click any place within the cue in-point field and then click the scroll bar to move it down. As the cue in-point increases, the duration decreases by the same amount.
- Select values with the left mouse button and then copy them using the middle mouse button.

If you press the Enter key after changing one of the values, the Media Control Panel verifies the new value. If the new value is valid, the Media Control Panel adjusts the other values to correspond to the new one, if necessary. If the new value is invalid, the Media Control Panel displays an error message in a dialog box.

If you do not press the Enter key after changing one of the values, the Media Control Panel does not check the validity of the new value. When you perform a function that uses the new values (for example, cueing the clip), the Media Control Panel then performs validity checking and displays an error message if a value is invalid.

Note: Changes that you make to the cue points take effect the next time that the clip is cued. They have no affect on a clip that is already cued or is playing.

Any changes that you make to the edit points or cue points are temporary. The changes cease to exist after the clip is unloaded unless you do one of the following:

- To make changes to the edit points permanent (that is, stored persistently with the clip), click the *Save Marks* button after you make the changes.
- To make changes to the cue points permanent, copy them to the edit points and then click the *Save Marks* button. The next time that the clip is loaded, the newly saved edit points are copied to the cue points.

Note: Only the last set of edit points are saved. That is, when you save the edit points, they replace the edit points that are currently stored with the clip.

Controlling a Video Deck

This section describes how to use the Deck Control window to control a video deck that is attached to the VCP-Recorder host. (For information on attaching a video deck to a VCP-Recorder server, see the *VCP-Recorder Installation and Administration Guide*.)

Note: To use the Deck Control window, you must specify a deck control port in addition to a video port when starting the Media Control Panel. (See “Starting the VCP-Recorder Media Control Panel” on page 16 for details.)

The following topics are discussed in this section:

- “Accessing the Deck Control Window” on page 33
- “About the Deck Control Window” on page 34
- “Recording From the Deck to a Clip” on page 37
- “Recording From a Clip to the Deck” on page 38

Accessing the Deck Control Window

To access the Deck Control window, choose View > Deck Control Panel from the menu bar of the Media Control Panel. The Deck Control window, shown in Figure 2-5, appears in a separate window.

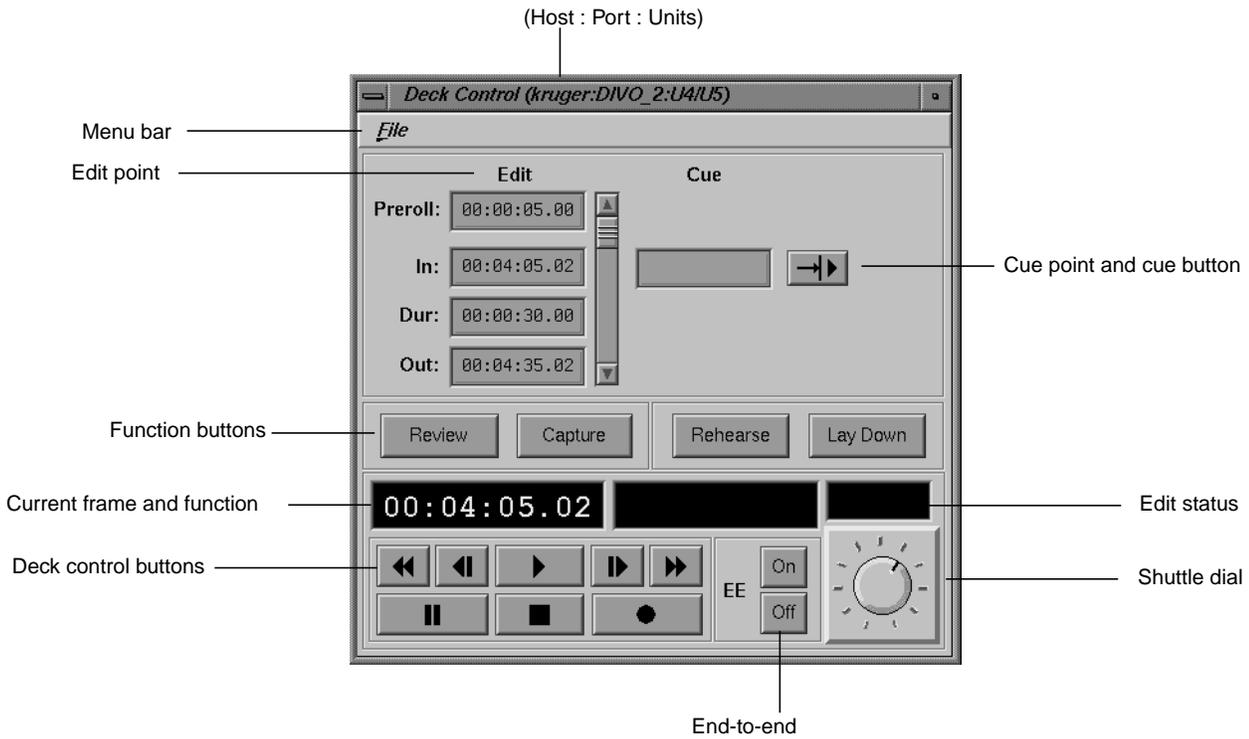


Figure 2-5 Deck Control Window

About the Deck Control Window

The File pulldown menu in the menu bar of the Deck Control window gives you access to the following options:

- Close, to close the Deck Control window
- Exit, to close both the Deck Control window and the Media Control Panel from which it was launched.

The following describes each of the displays and buttons in the Deck Control window:

- The *edit points* let you set the amount of time to preroll, the in-point, the duration, and the out-point. Each of these can be specified in the following format:

hh:mm:ss:ff

where *hh* is the hours, *mm* is the minutes, *ss* is the seconds, and *ff* is the frame number. In drop-frame mode, the final colon is replaced by a period:

hh:mm:ss.ff

See “Changing Cue Points and Edit Points” on page 32 for information about how to change the edit points.

- The *cue point* is the location at which you want to position the deck. The cue point can be specified in the following format:

hh:mm:ss:ff

where *hh* is the hours, *mm* is the minutes, *ss* is the seconds, and *ff* is the frame number. In drop-frame mode, the final colon is replaced by a period:

hh:mm:ss.ff

- The cue button (--> | >) tells VCP-Recorder to search the deck to the cue point and park it there.
- You can use the following deck control function buttons:
 - *Review*, to review the specified edit. The deck plays the selected portion of the tape, but VCP-Recorder does not record it.
 - *Capture*, to perform a frame-accurate capture from the selected portion of the tape to the VCP-Recorder *clip* currently loaded.
 - *Rehearse*, to review the specified edit. VCP-Recorder plays the selected portion of the clip, but the deck does not record it.
 - *Lay Down*, to perform a frame-accurate record from the selected portion of the clip to the tape.
- The current frame display is initialized to the cue in-point each time the clip is cued. As the clip is played or recorded, the display changes to indicate the current frame number.
- The function display shows the current function.

- The edit status display shows the status when performing an automated edit (that is, a capture, review, lay-down, or rehearse). The status may be one of the following:
 - CUE, when searching for the *cue point*
 - SYNC, when doing a preroll
 - LOCK, when the deck transport is locked
 - EDIT, when the edit is in progress
 - DONE, when the edit is complete
- The deck control buttons, which control playout and recording, correspond to standard VTR buttons. The following describes these buttons:
 - Fast-reverse (<<) moves the deck in reverse at a fast speed.
 - Jog backward (<|) jogs the deck backward by one frame. Each time you click the jog backward button, the deck jogs back one frame.

If the deck is playing when you click this button, the deck jogs backward one frame and then pauses. You have to click either the play button or the pause button to resume play.
 - Forward play (>) moves the deck in the forward direction. If the deck is not cued, it is cued before it begins playout.
 - Jog forward (|>) jogs the deck forward by one frame. Each time you click the jog forward button, the deck advances one frame.

If the deck is playing when you click this button, the deck jogs forward one frame and then pauses. You have to click either the play button or the pause button to resume play.
 - Fast-forward (>>) plays the deck forward at a fast speed.
 - Pause (| |) temporarily stops the deck from playing or recording. You have to click the play, record, or pause button to resume.
 - Stop (■) stops the deck and de-cues it. After the deck has been stopped, you must re-cue it to play it again.
 - Record (•) begins recording.

- The EE *On* and *Off* buttons turn the deck's end-to-end mode on and off, respectively. If you click the *On* button, the output displays the input signal instead of the signal from the deck. If you click the *Off* button, the output displays the signal from the deck.
- The shuttle dial lets you control the speed of the deck. To use the shuttle dial, use the mouse to point to the black notch on the dial. Press the left mouse button and keep it pressed while you turn the dial clockwise to increase the speed or counterclockwise to decrease it.

Note: The straight up position of the dial is zero, or pause.

Recording From the Deck to a Clip

To record from the deck to a *clip*, follow these steps:

1. Use the Media Control Panel to do the following:
 - Create a new clip. (See "Creating a New Clip" on page 31 for information about how to create a new clip.)
 - Change the cue in-point, if needed.

Note: The cue in-point in the Media Control Panel is the point in the clip to which you want the material captured.
 - If the Deck Control window is not displayed, choose View > Deck Control Panel in the menu bar of the Media Control Panel.
2. Use the Deck Control window to do the following:
 - Review the clip's *edit points* and change them if needed.

Note: The edit points in the Deck Control window identify the portion of the tape that you wish to capture and the duration of the capture.
 - To see what would be recorded without actually recording into the clip, click the *Review* button.
 - To record the clip from the deck, click the *Capture* button.

Recording From a Clip to the Deck

To record from a clip to the deck, follow these steps:

1. Use the Media Control Panel to do the following:
 - Load the clip from which you want to record.
 - Change the cue in-point, if needed.
Note: The cue in-point in the Media Control Panel is the point in the clip from which you want to record.
 - If the Deck Control window is not displayed, choose View > Deck Control Panel in the menu bar of the Media Control Panel.
2. Use the Deck Control window to do the following:
 - Review the edit points and change them if needed.
Note: The edit points in the Deck Control window identify the portion of the tape to which you want to record and the duration of the recording.
 - To see what would be recorded without actually recording onto the tape, click the *Rehearse* button.
 - To record from the clip to the deck, click the *Lay Down* button.

Monitoring the Status of VCP-Recorder

To monitor the status of VCP-Recorder, enter the following, either from the workstation on the VCP-Recorder server or from a workstation on which the VCP-Recorder tools software¹ has been installed:

```
% /usr/vtr/bin/mcstat [-v loglevel] [hostname]
```

where

- *loglevel* sets the severity level of the messages that are written on STDOUT.

Note: The *mcstat* program writes its log messages to the window from which it is invoked.

If this option is omitted, all messages with a severity level of Info and above are written on STDOUT. If this option is present, *loglevel*, which can be a positive or negative number, identifies the minimum level of the messages that are written to the log. (See Table 2-2 on page 18 for the definition of the log severity levels.)

- *hostname* is the name of the host on which VCP-Recorder is running. If this parameter is omitted, it is assumed that VCP-Recorder is running on the same host as the one from which the program is invoked.

The VCP-Recorder Unit Status Monitor appears in a separate window, which is shown in Figure 2-6. This monitor identifies each *port*, *unit*, type of connection and host, clip name (if one is loaded), current function and frame, and the status of unit. (See “About the Media Control Panel” on page 22 for more information.)



| Unit Status Monitor (007) | | | | | | |
|---------------------------|----|----------|-------------|---------|-------------|------|
| mvp | U4 | nvcv/007 | SeedsOfLife | PLAY | 00:00:00.00 | BUSY |
| | U5 | nvcv/007 | comm1 | STANDBY | | DONE |

Figure 2-6 Unit Status Monitor

¹ To run the Unit Status Monitor from a remote workstation, the *vcp_recorder_eoe.sw32.tools* subsystem must be installed on a workstation that has IRIX 6.2 or later. See the *VCP-Recorder Installation and Administration Guide* for more information.

See “About the Media Control Panel” on page 22 and “About the Deck Control Window” on page 34 for a description of the status information that is displayed by the Unit Status Monitor.

Note: You can also start the Unit Status Monitor by choosing Utilities > Status Monitor in the Media Control Panel menu bar.

Using the Clip Manager

This chapter describes how to use the Clip Manager graphical user interface (GUI) to manage VCP-Recorder *clips*, including how to get them from, and write them to, an *archive system*. The following topics are discussed:

- “Starting the Clip Manager” on page 42
- “Clip Manager Menus” on page 44
- “Obtaining Information About a Clip” on page 45
- “Renaming a Clip” on page 46
- “Deleting a Clip” on page 47
- “Setting the Protections for a Clip” on page 48
- “Using VCP-Recorder With an Archive System” on page 49
 - “Locating a Clip in an Attached Archive System” on page 50
 - “Bringing In a New Clip From an Attached Archive System” on page 51
 - “Updating a Clip” on page 52
 - “Writing a Clip to an Attached Archive System” on page 53

Starting the Clip Manager

To start the Clip Manager, enter the following, either from the workstation on the VCP-Recorder server or from a workstation on which the VCP-Recorder tools software¹ has been installed:

```
% /usr/vtr/bin/mcclips [-v loglevel] [hostname]
```

where

- *loglevel* sets the severity level of the messages that are written on STDOUT.

Table 3-1 shows the log severity levels and codes, which are listed in decreasing order of severity

Table 3-1 Log Severity Levels and Codes

| Priority | Log Level | Description |
|----------------|-----------|--|
| Emergency | -6 | Panic condition |
| Alert | -5 | Condition that should be corrected immediately, such as a corrupted system file |
| Critical | -4 | Critical condition that has system-wide impact, such as a hard device error; immediate action required |
| Error | -3 | Problem that needs correcting but does not require immediate action |
| Warning | -2 | Possible problem but could be a transient problem that corrects itself |
| Notice | -1 | Condition that might require attention, but is not an error condition |
| Info | 0 | Informational message |
| Debug <i>n</i> | <i>n</i> | Informational message that normally is of use to engineers for debugging; may be Debug1, Debug2, or Debug3, with Debug3 producing the most debugging information |

¹ To run the Clip Manager from a remote workstation, the `vcp_recorder_eoe.sw32.tools` subsystem must be installed on a workstation that has IRIX 6.2 later. See the *VCP-Recorder Installation and Administration Guide* for more information.

Note: The *mcclips* program writes its log messages to the window from which it is invoked.

If the *-v* option is omitted, all messages with a severity level of Info and above are written to the log. If this option is present, *loglevel*, which can be a positive or negative number, identifies the minimum level of the messages that are written to the log.

- *hostname* is the name of the host on which VCP-Recorder is running. If this parameter is omitted, it is assumed that VCP-Recorder is running on the same host as the one from which the program is invoked.

The VCP-Recorder Clip Manager is displayed in a separate window, which is shown in Figure 3-1. The window shows the clips that are in VCP-Recorder. For each clip, it shows the name, the duration, and the format. (If the duration contains an “*” it means that no material has been recorded in the clip.)

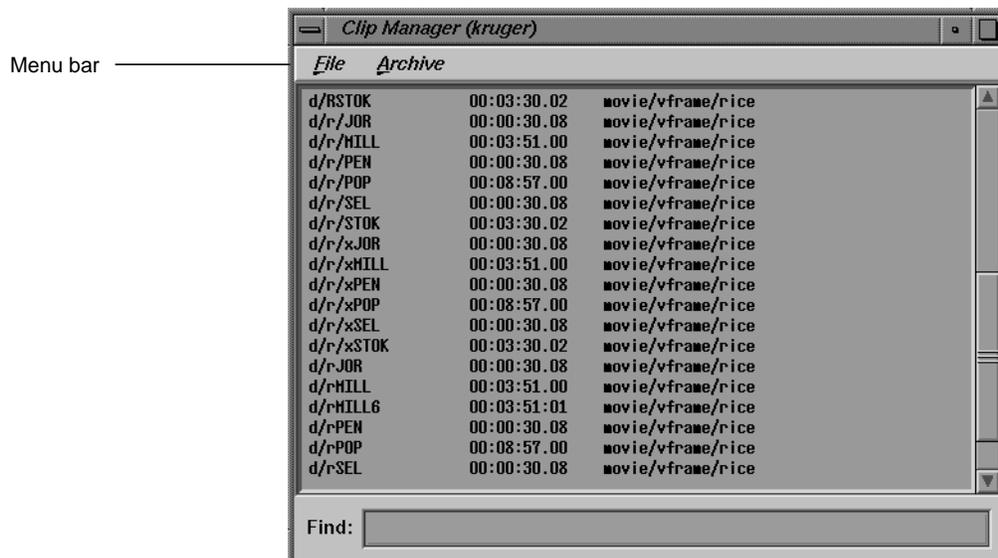


Figure 3-1 Clip Manager Window

Note: You can also start the Clip Manager by choosing Utilities > Clip Manager in the VCP-Recorder Media Control Panel menu bar.

Clip Manager Menus

The File pulldown menu in the menu bar of the Clip Manager window provides the following options:

- Info, to obtain information about a *clip*
- Rename, to rename a clip
- Delete, to delete a clip
- Protect, to set the protection levels for a clip
- Close, to close the Clip Manager window
- Exit, to close the Clip Manager window and exit the program

Except for the Close and Exit options, you must select a clip before requesting one of the options. You can select the clip by pointing to it with the mouse cursor and pressing the left mouse button.

You can also use the Find field to select a clip by entering any number of characters that match the values in either the name, the duration, or the format. For example, if you enter “xxx” in the Find field, the Clip Manager highlights the first clip that it finds that contains “xxx.” If you add “y” to the Find field, it searches for a clip that it finds that contains “xxy,” starting with the current clip.

Regardless of the method for selecting a clip, the selected clip becomes highlighted.

Note: You can access the clip-related functions in the File pulldown menu by selecting a clip and then pressing the right mouse button. You can also access the Info option by double-clicking the clip in the Clip Manager window.

The Archive pulldown menu in the menu bar provides the following options:

- Find, to get information about a clip in the *archive system*
- Get New, to bring a new clip from the archive system into VCP-Recorder
- Get, update an existing clip from the archive system
- Put, to write a clip to the archive system

Note: You can access the functions in this option menu by pressing the right mouse button and then selecting the Archive option in the pop-up menu.

Obtaining Information About a Clip

To obtain information about a *clip*, follow these steps:

1. Select the clip in the Clip Manager window.
2. Choose File > Info from the menu bar. The clip information window, as shown in Figure 3-2, appears.

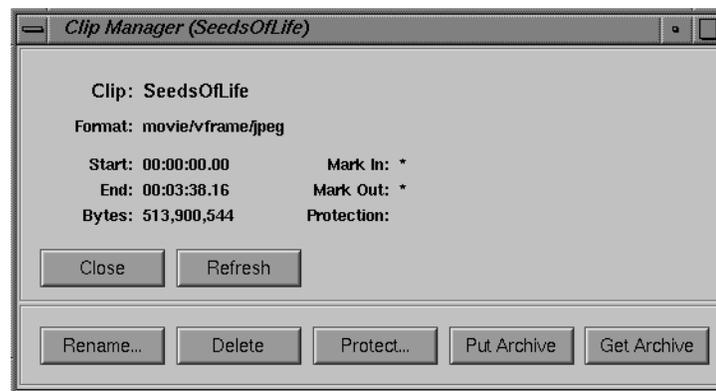


Figure 3-2 The Clip Information Window

This window shows the in- and out-points of the clip, the edit points, the size of the clip, and the protection. An asterisk (“*”) indicates that the value is not set. For example, Figure 3-2 indicates that the edit points are not set.

3. To refresh the information in this window, click the *Refresh* button.

Renaming a Clip

To rename a clip, follow these steps:

1. Do one of the following:
 - Select the clip in the Clip Manager window and then choose File > Rename from the menu bar.
 - If the clip information window for the clip is displayed, click the *Rename...* button in the information window.

The Rename Clip window, as shown in Figure 3-3, appears.

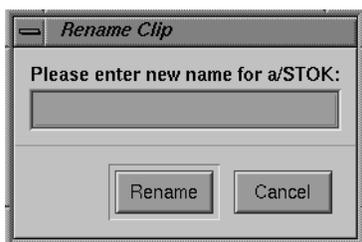


Figure 3-3 Rename Clip Window

2. Enter the new name of the clip.
3. Click the *Rename* button.

Deleting a Clip

To delete a clip, follow these steps:

1. Do one of the following:
 - Select the clip in the Clip Manager window and then choose File > Delete from the menu bar.
 - If the clip information window for the clip is displayed, click the *Delete* button in the information window.

The Delete Clip window, as shown in Figure 3-4, appears.

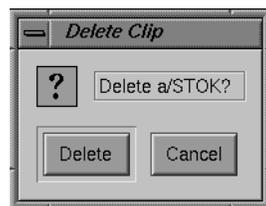


Figure 3-4 Delete Clip Window

2. Click the *Delete* button to delete the clip.

Setting the Protections for a Clip

By default, a clip can be deleted, recorded into, and renamed, and its *edit point* attributes can be changed. Therefore, if you want a clip to be protected from any of these changes, you must set the protections for that clip. This would be especially useful when multiple applications and/or *automation controllers* are using the VCP-Recorder.

To set the protections for a clip, follow these steps:

1. Do one of the following:
 - Select the clip in the Clip Manager window and then choose File > Protect from the menu bar.
 - If the clip information window for the clip is displayed, click the *Protect...* button in the information window.

The Set Protections window, as shown in Figure 3-5, appears.



Figure 3-5 Set Protections Window

This window identifies the protections that can be set. Each protection is preceded by a check box. If there's a red check in the check box, the corresponding protection is selected. If there is not a red check, the corresponding protection is not selected.

Note: The default is that a clip has no protections set. Therefore, you must set protections if you want the clip to have any.

2. Click the appropriate check boxes to indicate the protections that you want for this clip.

Note: Each check box acts like a toggle switch. Click the check box once to select that protection. Click the check box a second time to deselect it.

3. Click the *OK* button to have the protections take effect.

Using VCP-Recorder With an Archive System

When VCP-Recorder is configured to be used with MediaHub *archive systems*, you can use the VCP-Recorder GUI to

- locate MediaHub *clips*
- bring a MediaHub clip into VCP-Recorder
- update a VCP-Recorder clip by getting its content from MediaHub
- store a VCP-Recorder clip in MediaHub

See Appendix A, “Using a MediaHub Archive System,” for information about the structure of MediaHub clips that can be brought into VCP-Recorder or that were stored by VCP-Recorder in the archive system. See the *MediaHub User’s Guide* for information about MediaHub.

Locating a Clip in an Attached Archive System

To locate a clip in an attached *archive system*, follow these steps:

1. Choose Archive > Find from the menu bar. The Find Clip in Archive window, as shown in Figure 3-6, appears.



Figure 3-6 Find Clip in Archive Window

2. Enter the name of the clip. This corresponds to the value of the MediaHub *asset's ClipId* attribute.
3. Click the *Find* button.

Note: This feature is not fully implemented in the graphical user interface. To determine whether the clip was found, you must look in the VCP-Recorder log. (See the *VCP-Recorder Installation and Administration Guide* for information.)

Bringing In a New Clip From an Attached Archive System

To bring in a new clip from an attached *archive system*, follow these steps:

1. Choose Archive > Get New from the menu bar. The Get Clip from Archive window, as shown in Figure 3-7, appears.



Figure 3-7 Get Clip from Archive Window: Getting a New Clip

2. Enter the name of the clip, which is the value of the MediaHub asset's **ClipId** attribute.
3. Click the *Get* button.

Note: The clip is brought in from the first archive system in which it is found. That is, if the clip exists in more than one attached archive system, it is brought in from the first one. The clip can be brought in from an archive system other than the first one by using the MVCP command *AGET*.

Updating a Clip

To update a clip by bringing in new content from an attached *archive system*, follow these steps:

1. Do one of the following:
 - Select the clip in the Clip Manager window and then choose the Archive > Get option from the menu bar.
 - If the clip information window for the clip is displayed, click the *Get Archive* button in the information window.

The Get Clip from Archive window, as shown in Figure 3-8, appears.



Figure 3-8 Get Clip from Archive Window: Updating an Existing Clip

2. Click the *Get* button to update the clip from the archive.

Note: The clip is updated from the first archive system in which the clip is found. That is, if the clip exists in more than one attached archive system, it is updated from the first one. The clip can be updated from an archive system other than the first one by using the MVCP command *AGET*.

Writing a Clip to an Attached Archive System

To write a clip to an attached *archive system*, follow these steps:

1. Do one of the following:
 - Select the clip in the Clip Manager window and then choose Archive > Put from the menu bar.
 - If the clip information window for the clip is displayed, click the *Put Archive* button in the information window.

The Put Clip to Archive window, as shown in Figure 3-9, appears.



Figure 3-9 Put Clip to Archive Window

2. Click the *Put* button to write the clip to the archive system.

Note: The clip is written to the first attached archive system. That is, if there is more than one attached archive system, the clip is written to the first one. The clip can be put to an archive system other than the first one by using the MVCP command *APUT*.

MVCP Reference

This chapter contains detailed information about using the Silicon Graphics Multiple-Unit Video Computer Protocol (MVCP) command set. MVCP enables full-featured control of VCP-Recorder using a TCP/IP stream connection.

The MVCP commands are used by applications written specifically for use with VCP-Recorder. The commands can also be used by establishing an interactive VCP-Recorder connection, as discussed in “Establishing an Interactive MVCP Connection” on page 56. This is useful for interactive testing of features of VCP-Recorder.

The following topics are discussed in this chapter:

- “Using the MVCP Commands” on page 55
- “MVCP Global Commands” on page 66
- “MVCP Unit Commands” on page 97

Using the MVCP Commands

This section contains general information about using the MVCP commands. The following topics are discussed:

- “Establishing an Interactive MVCP Connection” on page 56
- “Rules for Using MVCP Commands” on page 56
- “About MVCP Timecodes” on page 57
- “MVCP Responses” on page 59
- “Synchronization of Unit Commands and Responses” on page 62
- “Command Timing” on page 65

Note: MVCP is similar in operation to that of *ftpd*, with the exceptions that are noted in this section.

Establishing an Interactive MVCP Connection

To establish an interactive MVCP connection, use telnet to establish a control connection to the MVCP port on the VCP-Recorder host.

The following example establishes an interactive connection to the MVCP port (which is usually port 5250) on the VCP-Recorder host “originserver.” The MVCP *PLS* (List Ports) command is run and then the *BYE* command is used to terminate the control connection:

```
% telnet originserver 5250
Trying 133.144.166.34...
Connected to originserver.abc.com.
Escape character is '^]'.
100 VTR Ready
PLS
201 OK
vlan_1 BOTH "VLAN Deck Control" DECK
DIVO_0 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_1 BOTH "SGI XT-DIVO Digital Video Option" VID
...
BYE
200 Goodbye
Connection closed by foreign host.
%
```

Rules for Using MVCP Commands

The following rules apply to the use of the MVCP commands:

- The commands are case-sensitive and must be entered exactly as shown in this chapter.
- All MVCP command arguments are space-delimited. Command arguments that contain spaces should be surrounded by double quotation marks.
- Each command must end with a carriage return and line feed.
- The command set is divided into two subsets. One subset contains the global commands, which are those commands that do not pertain to a specific *unit*. The other subset contains the unit commands, which are those commands that pertain to a specific unit. (See “MVCP Global Commands” on page 66 and “MVCP Unit Commands” on page 97 for detailed information about these command subsets.)

- MVCP commands that pertain to a specific *unit* contain the unit name as the first argument of the request. Global commands do not include a unit name. For example, the following is a global command that requests information about the *clip* named “NA40125D.” This command does not have a unit name:

```
CINF NA40125D           (get clip information)
```

The following is a unit command that loads the clip named “NA40125D” for output into the unit identified by “U1”:

```
LOAD U1 NA40125D OUT    (load clip into unit U1)
```

- MVCP responds to each command with a response header line, which is terminated with a carriage return and line feed. All MVCP responses are space-delimited. Response arguments that contain spaces are surrounded by double quotation marks. (See “MVCP Responses” on page 59 for information).

Note: Future enhancements to MVCP may result in additional arguments being added to commands or responses. The additional arguments will be added in such a way that the commands or responses will be backward-compatible with the existing commands or responses, so that existing applications will continue to work correctly.

About MVCP Timecodes

Timecodes are used in MVCP commands to indicate

- a clip’s frame number and edit points (see individual commands described in “MVCP Global Commands” on page 66 and “MVCP Unit Commands” on page 97)
- the time of day (see “STOD (Set System Time)” on page 93)
- the time at which a command should be executed (see “Command Timing” on page 65)

These MVCP timecodes are specified in the following 24-hour format:

```
[[hh:]mm:]ss:]ff
```

where *hh* is the hours, *mm* is the minutes, *ss* is the seconds, and *ff* is the frame number. In drop-frame mode, the final colon is replaced by a period (*hh:mm:ss.ff*).

When a frame number is specified in a timecode, a frame rate is used to determine the equivalent time. For example, if the frame rate is 30 frames per second, 00:00:00.15 (15 frames) refers to 0.5 seconds. If the frame rate is 25 frames per second, 00:00:00.15 refers to 0.6 seconds.

Note: When specifying frame numbers and edit points, the frame rate associated with the clip is used. When specifying the time at which a command is executed, the frame rate set by the *FRAT* command is used; the default frame rate (29.97) is used for command timing if the *FRAT* command has not been run.

Examples of timecodes are shown in Table 4-1.

Table 4-1 MVCP Timecode Examples

| Timecode | Description |
|-------------|--|
| 15 | equivalent to 00:00:00:15; actual time depends on frame rate frame numbers and edit points example: if clip frame rate is 25 frames per second, 00:00:00:15 is 0.6 seconds command timing example: if frame rate is 25 frames per second, 00:00:00:15 is 0.6 seconds past midnight |
| 15:00 | equivalent to 00:00:15:00 frame numbers and edit points: 15 seconds command timing: 15 seconds past midnight |
| 15:00.00 | equivalent to 00:15:00.00 frame numbers and edit points: 15 minutes command timing: 15 minutes past midnight |
| 15:00:00.00 | frame numbers and edit points: 15 hours command timing: 15 hours past midnight (3 p.m.) |
| 10:00:00:00 | frame numbers and edit points: 10 hours command timing: 10 hours past midnight (10 a.m.) |
| 00:00:30:15 | actual time depends on frame rate frame numbers and edit points example: if clip frame rate is 30 frames per second, 00:00:30:15 is 30.5 seconds command timing example: if frame rate is 30 frames per second, 00:00:30:15 is 30.5 seconds past midnight |

The following examples, which are all equivalent, cue the clip in unit U1 at 15 seconds:

```
CUE U1 00:00:15.00
CUE U1 00:15.00
CUE U1 15.00
```

The following example requests that the *PLAY* command be executed at 30 minutes after midnight:

```
@30:00:00 PLAY U1
```

MVCP Responses

The MVCP processor responds to all commands with a single response header line. Some commands also return either a single response data line or multiple response data lines. When multiple data lines are returned, they are terminated by a single blank line.

The response header line has the following format:

```
xxx message_text
```

where

- *xxx* is a three-digit decimal response code.
- *message_text* is a textual message describing the response.

This response header line may be followed by one or more data lines, depending on the response code.

Each of the following is an example of a response header line:

```
200 OK
410 Invalid clip time
500 Server error
```

The following is an example of a response header line that is followed by two data lines:

```
ULS
201 OK
U1 mvcp/originserver mvp BOTH * DONE IDLE * 0 *
U2 mvcp/originserver mvp BOTH * DONE IDLE * 0 *
```

Note: In this example, a blank line follows the second data line.

Caution: Additional fields may be added as future enhancements to response lines. The additional fields will be added in such a way that the responses will be backward-compatible so that existing applications will continue to work correctly.

The response codes are divided into the following categories:

- 2xx, successful command execution
- 4xx, command format or setup errors
- 5xx, command execution errors

2xx Response Codes

A response header line with a 2xx response code indicates the successful completion of the command. Each 2xx response header line may be followed by one or more data lines, according to the following:

- 200 response header lines are not followed by any data lines.
- 201 response header lines are followed by more than one data line. The last data line is followed by a blank line (that is, a line with just a carriage return and line feed).
- 202 response header lines are followed by a single data line.

4xx Response Codes

The 4xx format error codes include the following:

- 401 Port name missing
- 402 Unit name missing
- 403 Port not found
- 404 Access mode missing or invalid
- 405 Clip name missing
- 406 Command not supported
- 407 Load mode missing or invalid
- 408 <not used>
- 409 <not used>
- 410 <not used>
- 411 Event function missing or invalid
- 412 Clock time missing or invalid
- 413 Numeric argument missing or invalid
- 414 Invalid direction
- 415 <not used>
- 416 Insertion id missing or invalid
- 417 Event insertion failed
- 418 Invalid speed
- 419 Invalid number of passes
- 420 Must specify start time
- 421 Bad create flag
- 422 Invalid command sync
- 423 Invalid protection
- 424 Missing or invalid sort order
- 425 Missing or invalid time
- 426 Clip monitor not active
- 427 Invalid parameter format
- 428 Node type missing or invalid
- 429 Invalid event type
- 432 Too many units specified
- 433 Invalid timestamp format
- 434 Frame-rate missing
- 435 Filename missing
- 436 Still-frame mode invalid
- 437 Cannot specify both in and out as relative

5xx Response Codes

The 5xx execution error codes include the following:

500 General error (message varies)
510 General error (message varies)
511 General error (message varies)

Synchronization of Unit Commands and Responses

VCP-Recorder *unit* commands execute asynchronously with the control processors that run them. By default, when a command is issued to a unit, if that command has no format or setup errors, the command queue for that unit is flushed and the new command is placed on the command queue. In other words, the new command preempts all other commands issued to the unit, even if those commands have not completed their execution. Also by default, the MVCP control processor returns its response as soon as the command has been parsed for command format and setup errors and has been placed on the unit's command queue.

This section describes how to modify this default behavior. The following topics are discussed:

- “Changing the Synchronization Mode” on page 63
- “Changing Command Sequencing” on page 64

Changing the Synchronization Mode

The following are the available synchronization modes:

- ASYN, which is async mode. This is the default mode, which behaves as described. That is, the control processor responds as soon as a command is queued.
- SYNI, which is sync-init mode. This mode delays the response to the command until the *unit* has reached the initialization state on the command. This does not mean that the unit has begun to process the command, and in fact it may have to wait for a shared resource (such as the media *port*) to become available before it can continue.
- SYN R, which is sync-run mode. This mode delays the response to the command until the unit has reached the “running” state on the command. This means that the unit is actively processing the command. For instance, the running state for a *PLAY* command means that video is actually being played.
- SYNC, which is sync-complete mode. This mode delays the response to the command until the unit has reached the “complete” state on the command. This means that the unit does no further processing for this command. If the command is *CUE*, the complete state is reached when the unit has been fully cued and is ready to begin playback. If the command is *PLAY*, the complete state is reached when playback has finished.

If no error occurs in the unit before the desired sync state has been reached, the response code is 200, 201, or 202, as appropriate. If an error occurs before the unit reaches the desired state, a 5XX error response code is returned.

The default command synchronization mode for a unit can be changed using the *USYN* command. In addition, the default can be overridden on an individual command basis by prepending a forward slash (/) and the sync mode to the unit command. This is shown in the following example, which specifies that the *REC* command should be executed in SYNC mode:

```
/SYNC REC U1
```

Note: When a unit is first created, its default sync mode is set to ASYN. This default synchronization mode for a unit can be changed by using the unit command *SSYN*. The mode specified in this command is used for all unit commands that do not specify an explicit sync mode.

Changing Command Sequencing

Command sequencing refers to the way in which consecutive commands are issued to, and processed by, the target *unit*. There are two variables that determine how a command is issued and processed: the preemption mode and the queueing mode.

The preemption mode determines whether a command preempts the previous command or waits for the previous command to complete before being executed by the unit. The available preemption modes are immediate (IMM) and sequential (SEQ). An immediate command preempts the previous command, while a sequential command is not executed until the previous command completes on that unit.

The available queueing modes are PRE (prepend), APP (append), and FLSH (flush). These queueing modes determine whether a command is placed at the beginning of the unit command queue for a unit (making it the next command to be executed), placed at the end of the command queue (making it the last command), or whether the command queue is flushed before adding the new command.

The default command sequencing is flush-immediate, meaning that when an MVCP command is issued, the unit command queue is flushed and the command preempts whatever command is currently being executed by the unit.

The default command sequencing mode for a unit may be changed using the *USYN* command.

Note: If you use the *USYN* command to change the command sequencing mode for a unit, the change applies only to that unit. The change does not apply to other units that may be active.

The default can be overridden on an individual command basis by including a forward slash (/) and the sequencing mode at the beginning of the unit command. For example, the command *PLAY U5* uses the default sequencing mode, while the command */APP PLAY U5* specifies that the command should be appended to the unit's command queue.

The default preemption and sequencing modes are overridden individually. For example, */APP /SEQ PLAY U5* places the *PLAY* command at the end of the command queue, and the command does not begin executing until the previous command has completed.

Three common sequencing combinations can be abbreviated:

- `/SEQA` is equivalent to `/APP /SEQ`
- `/IMMP` is equivalent to `/IMM /PRE`
- `/IMMF` is equivalent to `/IMM /FLSH`

For example, `/APP /SEQ PLAY U5` is equivalent to `/SEQA PLAY U5`.

Command Timing

More precise control of command execution can be accomplished by specifying the time at which a given command should begin execution.

Note: This feature is supported only for *unit* transport commands, such as `PLAY`, `REC`, and `STOP`. Command timing is ignored on all other commands.

To specify a command time, include an at-sign (@) followed by the desired wall-clock time at the beginning of the command line. Two formats for specifying time are supported:

- The timecode form (hh:mm:ss:ff, or hh:mm:ss:ff in drop-frame mode) can be used to specify the time relative to midnight local time; it is calculated using the current frame rate. (The frame rate defaults to 29.97 and can be set with `FRAT`.) For example, 14:14:00:00 specifies that the command should start at 2:14 p.m. local time.

See “About MVCP Timecodes” on page 57 for information about the timecode format.

- The time-of-day form, `SEC.USEC`, can be used to specify the time using a more traditional UNIX timebase. `SEC` is the number of seconds since the standard Epoch, Jan 1, 1970. `USEC` is microseconds.

The following example loads a clip into a unit, cues the clip for playout, and then plays it at 5:30 p.m. The CUE and

```
LOAD U1 a/COMM OUT
/SEQA CUE U1
/SEQA @17:30:00:00 PLAY U1
```

Note: The command timing feature is intended to provide timed control over starting and stopping playback and recording. It is not intended for use in long-term scheduling because delays of more than a few seconds are possible.

MVCP Global Commands

Table 4-2 identifies the global MVCP commands that are documented in this section.

Table 4-2 MVCP Global Commands

| Command | Function |
|---------|---|
| AFND | Locate a clip in an available archive system |
| AFNG | Locate a pending or in-progress get-from-archive command for a clip |
| AGET | Retrieve a clip from an archive system |
| ALSG | List the get-from-archive operations that are waiting or being processed |
| ALSP | List the put-to-archive operations that are waiting or being processed |
| APUT | Store a clip to an attached archive system |
| ARM | Delete a clip from an attached archive system |
| ARMG | Cancel all pending or in-progress get-from-archive operations |
| BYE | Terminate the current control connection |
| CADD | Add a clip that has been inserted into the clip cache through an external means |
| CCHP | Add specified protections to, or remove them from, a clip |
| CCP | Create a new clip by copying it from an existing clip |
| CCST | Get the current status of the VCP-Recorder Clip Cache |
| CEDP | Set the edit points for a clip |
| CGP | Get clip protections |
| CIMG | Copy an image to a file |
| CINF | Get information about a clip |
| CLN | Create a new clip that shares the clip media content of another clip |
| CLS | Get a list of all the clips in the VCP-Recorder clip cache |
| CLSA | List the clips that have been added to the clip cache since the last time the CLSA command was issued or since the CMON command that created this clip monitor was executed |

Table 4-2 (continued) MVCP Global Commands

| Command | Function |
|----------------|---|
| CLSR | List the clips that have been removed from the clip cache since the last time the CLSR command was issued or since the CMON command that created this clip monitor was executed |
| CMIN | Get the number of clips that have been added to, or removed from, the clip cache |
| CMON | Initiate monitoring of the clip cache |
| CMV | Rename a clip |
| CRM | Delete a clip from the clip cache |
| CRMA | Delete all clips currently residing in the clip cache |
| ERR | Get the code and description of the last global error that occurred for a controller |
| FRAT | Set the frame rate used in translating timecodes for command timing |
| GTOD | Get the current system time |
| MON | Place control connections into event monitoring mode |
| PLS | Get the list of supported media ports |
| SORD | Set the sort order that's used when lists are returned |
| STLS | List statistics |
| STOD | Set the system time |
| STST | Calculate various statistics |
| STZ | Reset statistics |
| UADD | Create a new unit |
| ULS | Get a list of the existing VCP-Recorder units |

AFND (Find Clip in Archive)

To locate a *clip* in an attached *archive system*, use the following command:

```
AFND clip-name [archive-host]
```

where

- *clip-name* is the name of the clip to be located.
- *archive-host* is the hostname of the attached archive system that should be searched. If not provided, all attached archive systems are searched.

The *AFND* command attempts to locate the clip specified by *clip-name* in an attached archive system. If the command is successful, the response code is 201, and a response line is returned for the first archive system in which the clip is found. The response line is in the following format:

```
archive-type archive-host
```

where

- *archive-type* is the type of archive system.
- *archive-host* is the specific hostname of the archive system containing the clip.

For example:

```
AFND NA1001  
201 OK  
mediahub mhhost
```

Note: VCP-Recorder supports multiple MediaHub archive systems. If a specific archive system is not provided on the command, only the first archive host that has the requested clip name is returned, regardless of how many attached archive systems have a clip with that name. To find all the archive systems that have the clip, execute the *AFND* command multiple times, specifying a different archive hostname each time.

AFNG (Find Archive Get)

To locate a pending or in-progress get-from-archive command for a *clip*, use the following command:

```
AFNG clip-name
```

where *clip-name* is the name of the clip in VCP-Recorder.

The *AFNG* command attempts to locate a pending or in-progress get-from-archive command for the clip specified by *clip-name*. If the archive get command is found, the response code is 202 and a single response line is returned in the following format:

```
clip atype ahost requester requested started eta
```

where

- *clip* is the name of the clip.
- *atype* is the type of *archive system*.
- *ahost* is the specific hostname of the archive system.
- *requester* is the name of the controller requesting the clip.
- *requested* is the time the retrieval was requested in ISO 8601 format.
- *started* is the time the retrieval was started in ISO 8601 format.
- *eta* is the time the retrieval is expected to complete in ISO 8601 format.

AGET (Get From Archive)

To retrieve a *clip* from an *archive system*, use the following command:

```
AGET clip-name [archive-clip-name [archive-host]]
```

where

- *clip-name* is the clip name in the VCP-Recorder *clip cache*.
- *archive-clip-name* is the clip name in the archive system. If *archive-clip-name* is not specified, or the default *archive-clip-name* (“*”) is specified, *clip-name* is used.
archive-clip-name must be specified if either an archive host is specified or the clip name in the archive system is different from the clip name in VCP-Recorder.
- *archive-host* is the name of the archive host. If *archive-host* is specified, VCP-Recorder attempts to retrieve the clip from only that specific archive. Otherwise, the clip is retrieved from the first archive in which it can be located.

The following example gets a clip named “NA1002” from the archive host named “mhhost.” The clip name will be “NA1002” in the VCP-Recorder clip cache:

```
AGET NA1002 NA1002 mhhost
```

This command queues a get-from-archive operation to retrieve the clip specified by *clip-name* from the specified attached archive system. If a get-from-archive operation is pending for the clip, *AGET* does not add a new one.

If there are no format or setup errors in the command, *AGET* always returns immediately with a response code of 200.

Note: Archive operations are processed in order by the VCP-Recorder Archive Manager, which can process a configurable number of concurrent operations for each attached archive system.

If the clip cannot be successfully retrieved from the archive, the failure is logged in the VCP-Recorder status log. In addition, a clip monitor or event monitor treats the failing clip as having been removed from the clip cache.

ALSG (List Archive Gets)

To list the get-from-archive operations that are waiting or being processed, use the following command:

```
ALSG
```

This command lists the get-from-archive operations that are waiting or being processed. The response code is 201, and a single response line is returned for each archive request in the following format:

```
clip atype ahost requester requested started eta
```

where

- *clip* is the name of the *clip*.
- *atype* is the type of *archive system*.
- *ahost* is the specific hostname of the archive system.
- *requester* is the name of the controller requesting the clip.
- *requested* is the time the retrieval was requested in ISO 8601 format.
- *started* is the time the retrieval was started in ISO 8601 format.
- *eta* is the time the retrieval is expected to completed in ISO 8601 format.

ALSP (List Archive Puts)

To list the put-to-archive operations that are waiting or being processed, use the following command:

```
ALSP
```

This command lists the put-to-archive operations that are waiting or being processed. The response code is 201, and a single response line is returned for each archive request in the following format:

```
clip atype ahost requester requested started eta
```

where

- *clip* is the name of the *clip*.
- *atype* is the type of *archive system*.
- *ahost* is the specific hostname of the archive system.
- *requester* is the name of the controller requesting the clip.
- *requested* is the time the retrieval was requested in ISO 8601 format.
- *started* is the time the retrieval was started in ISO 8601 format.
- *eta* is the time the retrieval is expected to complete in ISO 8601 format.

Note: *atype* and *requester* can be null.

APUT (Put to Archive)

To store a *clip* to an attached *archive system*, use the following command:

```
APUT clip-name [archive-clip-name [archive-host]]
```

where

- *clip-name* is the clip name in the VCP-Recorder *clip cache*.
- *archive-clip-name* is the clip name in the archive system. If *archive-clip-name* is not specified, or the default *archive-clip-name* (“*”) is specified, *clip-name* is used.
archive-clip-name must be specified if either an archive host is specified or the clip name in the archive system is different from the clip name in VCP-Recorder.
- *archive-host* is the name of the archive host. If *archive-host* is specified, the clip is stored into that specific archive. Otherwise, the clip is stored into the default archive.

The following example stores a clip named “NA1004” in the archive host named “mhhost.” The clip name will be “NA1002” in the archive host:

```
APUT NA1004 NA1002 mhhost
```

The *APUT* (Put to Archive) command queues a put-to-archive operation to store the clip specified by *clip-name* to an attached archive system. If a put-to-archive operation is pending for the clip, *APUT* does not add a new one.

If the specified clip does not exist, *APUT* returns an appropriate error response code. Otherwise, *APUT* immediately returns response code 200.

Note: Archive operations are processed in order by the VCP-Recorder Archive Manager, which can process a configurable number of concurrent operations for each attached archive system.

If the clip cannot be successfully stored into the archive, the failure is logged in the VCP-Recorder status log.

ARM (Delete From Archive)

To delete a *clip* from an attached *archive system*, use the following command:

```
ARM archive-clip-name [archive-host]
```

where

- *archive-clip-name* is the name of the clip to be deleted.
- *archive-host* is the name of the archive host. If it is omitted, the first archive system is used.

This command queues a delete-from-archive operation to delete the clip specified by *archive-clip-name* from an attached archive system. (If a delete-from-archive operation is pending for the clip, *ARM* does not add a new one.)

If there are no format or setup errors in the command, *ARM* always returns immediately with a response code 200.

Archive operations are processed in order by VCP-Recorder, which can process a configurable number of concurrent operations for each attached archive system.

ARMG (Cancel Archive Gets)

To cancel all pending or in-progress get-from-archive operations, use the following command:

```
ARMG
```

This command cancels all pending or in-progress get-from-archive operations.

If there are no format or setup errors in the command, *ARMG* always returns immediately with a response code 200.

BYE

To terminate the current control connection, use the following command:

```
BYE
```

CADD (Add Clip)

To add a *clip* that has been inserted into the *clip cache* through an external means, use the following command:

```
CADD clip-name [clip-format]
```

where

- *clip-name* is the name of the new clip.
- *clip-format* is the format of the clip. The format of the clip must be specified unless the clip already has the “clip.format” attribute attached.

This parameter can be one of the following:

- movie/vframe/uncompressed (uncompressed VCP-Recorder variable-frame format movies)
- movie/vframe/rice (compressed VCP-Recorder variable-frame format movies using *Rice* compression)
- movie/vframe/jpeg (motion JPEG)
- movie/stream/mpeg/mpeg2/program (MPEG-2 program stream)
- movie/stream/mpeg/mpeg2/transport (MPEG-2 transport stream)

The clip media file must be placed in the appropriate location in the clip cache before issuing the *CADD* command. If the clip format requires an index, the index file must also be placed in the appropriate index directory in */usr/vtr/index* before issuing the command. For example, if the media file is */usr/vtr/clips/news/current/stocks*, the index file would be stored in */usr/vtr/index/news/current/stocks*.

CCHP (Change Clip Protection)

To add specified protections to, or remove them from, a *clip*, use the following command:

```
CCHP clip-name {{+ | -}protection-type ...}
```

where

- *clip-name* is the name of the clip.
- *protection-type* is the type of protection. This parameter can be one of the following:
 - ATTR, which protects the clip from changes to the edit in-point and edit out-point attributes
 - MV, which protects the clip from renaming
 - REC, which protects the clip from recording
 - RM, which protects the clip from deletes

The plus sign (“+”) indicates that the protection should be added, and the minus sign (“-”) indicates that it should be removed.

Note: The default is that a clip has no protections set. Therefore, you must use this command if you want the clip to have any protections.

For example, the following command sets the protection on the clip named “NA1001” such that the clip cannot be deleted but it can have its attributes (that is, its edit in- and out-points) changed:

```
CCHP NA1001 +RM -ATTR
```

If the command is successful, the response code is 200.

CCP (Copy Clip)

To create a new *clip* by copying it from an existing clip, use the following command:

```
CCP clip-name new-clip-name
```

where

- *clip-name* is the name of the existing clip used to create the new clip.
- *new-clip-name* is the name of the clip that is created.

The clip is created with all the same attributes as the original clip. However, after it has been created, the newly created clip has no association with the original clip.

If the command is successful, the response code is 200.

CCST (Clip Cache Status)

To get the current status of the VCP-Recorder *clip cache*, use the following command:

```
CCST
```

This command returns a single response line (response code of 202) that is in the following format:

```
num-clips bytes-used bytes-available bytes-avail-contiguous
```

where

- *num-clips* is the number of clips resident in the clip cache.
- *bytes-used* is the number of bytes used on the disk system that holds the clip cache.
- *bytes-available* is the number of free bytes on the disk system that holds the clip cache.
- *bytes-avail-contiguous* is the number of free bytes available on the storage system that has the greatest amount of free space.

CEDP (Set Edit Point)

To set the edit points for a *clip*, use the following command:

```
CEDP clip-name in-point out-point
```

where

- *clip-name* is the clip for which the edit points are set.
- *in-point* is the edit in-point, specified in hh:mm:ss:ff format (hh:mm:ss.fff in drop-frame mode). Specifying "*" for *in-point* removes the respective edit point.
- *out-point* is the edit out-point, specified in hh:mm:ss:ff format (hh:mm:ss.fff in drop-frame mode). Specifying "*" for *out-point* removes the respective edit point.

This command sets the edit points to the specified values. The new edit points are not seen by any *unit* until the clip is loaded. If the clip is already loaded into a unit when this command is run, the original edit points still apply.

If the command is successful, the response code is 200.

CGP (Get Clip Protection)

To get the current protections for a *clip*, use the following command:

```
CGP clip-name
```

where *clip-name* is the name of the clip for which the protections are wanted.

If this command is successful, the response code is 202, and a single response line is returned. The response line contains a list of the protections currently enabled for the clip. The protections may be the following:

- ATTR (Attribute Protect)
- MV (Rename Protect)
- REC (Record Protect)
- RM (Delete Protect)

CIMG (Create Clip Image)

To copy an image from a *clip* to a file, use the following command:

```
CIMG clip-name timecode interleave filename [format]
```

where

- *clip-name* is the name of the clip.
- *timecode* is the timecode of the frame to be extracted (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).
- *interleave* specifies how to construct the image from the two fields of the frame: f1 (odd field only), f2 (even field only), f1f2 (both fields interleave), f1f1 (odd field line-doubled), f2f2 (even field line-doubled).
- *filename* is the name of the file.
- *format* specifies the image format to use. Possible values are “rice,” “rgb,” “yuv,” “jpeg,” and “tiff” (or others as supported by the Image Format Library). If format is not specified, the format is inferred from the filename extension (“.rice,” “.rgb,” “.yuv,” “.jpg,” “.tiff”).

The *CIMG* (Create Clip Image) command extracts the image data associated with the clips’s frame (specified by *timecode*) and writes the image to the file specified by *filename*.

CINF (Clip Info)

To get information about a *clip*, use the following command:

```
CINF clip-name
```

where *clip-name* is the name of the clip.

This command returns the attributes of the clip specified by *clip-name*. If the clip exists in the *clip cache*, the response code is 202 and a single response line is returned in the following format:

```
clip-name format size resident-size start end in-point out-point frame-rate
```

where

- *clip-name* is the name of the clip.
- *format* is the format of the clip. The format can be one of the following:
 - *movie/vframe/uncompressed* (uncompressed VCP-Recorder variable-frame format movies)
 - *movie/vframe/rice* (compressed VCP-Recorder variable-frame format movies using *Rice* compression)
 - *movie/vframe/jpeg* (motion JPEG)
 - *movie/stream/mpeg/mpeg2/program* (MPEG-2 program stream)
 - *movie/stream/mpeg/mpeg2/transport* (MPEG-2 transport stream)
- *size* is the size of the clip content in bytes.
- *resident-size* is the size of the content that is currently resident on the clip cache disk system.
- *start* is the timecode of the first frame of the clip (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).
- *end* is the timecode of the frame after the last frame of the clip (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).

- *in-point* is the current edit in-point of the clip (hh:mm:ss:ff, or hh:mm:ss.fff in drop-frame mode).
- *out-point* is the current edit out-point of the clip (hh:mm:ss:ff, or hh:mm:ss.fff in drop-frame mode).
- *frame-rate* is the frame rate of the clip in frames per second. 525-line timing is specified in its approximate form (29.97 frames per second).

CLN (Link Clip)

To create a new *clip* that shares the clip media content of another clip, use the following command:

```
CLN clip-name new-clip-name
```

where

- *clip-name* is the name of the existing clip whose content is shared by the new clip.
- *new-clip-name* is the name of the clip that is created.

The new clip starts with all the same attributes and protections as the original clip. However, the attributes of the new clip (such as its edit points) can be set independently of the original clip.

If the original clip is deleted, the new clip retains the content of the original clip. If both clips are deleted, the content is also deleted.

This command is useful for creating clips that refer to smaller segments of other clips.

If the command is successful, the response code is 200.

CLS (List Clips)

To get a list of all the *clips* in the VCP-Recorder *clip cache*, use the following command:

```
CLS
```

The response code for this command is 201. The response header line is followed by one response line for each clip in the cache. The format of each of those lines is:

```
clip-name format size resident-size start end in-point out-point frame-rate
```

where

- *clip-name* is the name of the clip.
- *format* is the format of the clip. The format can be one of the following:
 - *movie/vframe/uncompressed* (uncompressed VCP-Recorder variable-frame format movies)
 - *movie/vframe/rice* (compressed VCP-Recorder variable-frame format movies using *Rice* compression)
 - *movie/vframe/jpeg* (motion JPEG)
 - *movie/stream/mpeg/mpeg2/program* (MPEG-2 program stream)
 - *movie/stream/mpeg/mpeg2/transport* (MPEG-2 transport stream)
- *size* is the size of the clip content in bytes.
- *resident-size* is the size of the content that is currently resident on the *clip cache* disk system.
- *start* is the timecode of the first frame of the clip (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).
- *end* is the timecode of the frame after the last frame of the clip (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).
- *in-point* is the current edit in-point of the clip (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).
- *out-point* is the current edit out-point of the clip (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).
- *frame-rate* is the frame rate of the clip in frames per second. 525-line timing is specified in its approximate form (29.97 frames per second).

CLSA (List Added Clips)

To list the *clips* that have been added to the *clip cache* since the last time the *CLSA* command was issued (or since the *CMON* command that created this clip monitor was executed), use the following command:

```
CLSA
```

Before the *CLSA* command may be issued by a controller, the *CMON* (Clip Monitor) command must be issued to initiate monitoring of the clip cache for this controller.

If this command is successful, the response code is 201. One response line is returned for each newly added clip, listing just the clip's name. A blank line terminates the list of clips.

CLSR (List Removed Clips)

To list the *clips* that have been removed from the *clip cache* since the last time the *CLSR* command was issued or since the *CMON* command that created this clip monitor was executed, use the following command:

```
CLSR
```

Before this command may be issued by a controller, the *CMON* (Clip Monitor) command must be issued to initiate monitoring of the clip cache for this controller.

If this command is successful, the response code is 201. One response line is returned for each newly removed clip, listing just the clip's name. A blank line terminates the list of clips.

CMIN (Clip Monitor Info)

To get the number of *clips* that have been added to or removed from the *clip cache* since the last time the *CLSA* and *CLSR* commands were issued, use the following command:

```
CMIN
```

This command returns the same counts that would be returned if the *CLSA* and *CLSR* commands were issued.

The response code for this command is 200 and a single response line is returned in the following format:

```
num-clips-added num-clips-deleted
```

where

- *num-clips-added* is the number of clips that have been added since the last time the *CLSA* command was issued.
- *num-clips-deleted* is the number of clips that have been removed since the last time the *CLSR* command was issued.

CMON (Clip Monitor)

To initiate monitoring of the *clip cache*, use the following command:

```
CMON
```

This command initiates the monitoring of the clip cache for the *CLSA* and *CLSR* commands, which are used to get a list of the clips that have been added or removed, respectively.

If the *CMON* command is issued for a second or subsequent times the current list of added and removed clips is discarded and monitoring is initialized again.

The response code to this command is 200.

CMV (Rename Clip)

To rename a *clip*, use the following command:

```
CMV clip-name new-clip-name
```

where *clip-name* is the existing name of the clip and *new-clip-name* is its new name.

If the command is successful, the response code is 200.

CRM (Delete Clip)

To delete a *clip* from the *clip cache*, use the following command:

```
CRM clip-name
```

where *clip-name* is the name of the clip to be deleted. If any *units* currently have the clip loaded, actual deletion of the clip is deferred until all units have unloaded the clip.

If the command is successful, the response code is 200.

CRMA (Delete All Clips)

To delete all *clips* currently residing in the *clip cache*, use the following command:

```
CRMA
```

Clips that are currently being retrieved from an *archive system* may or may not be deleted depending on the progress of the retrieval.

If the command is successful, the response code is 200.

ERR (Error)

To get the code and description of the last global error that occurred for a controller, use the following command:

```
ERR
```

This command returns the code and description of the last global error that occurred for this controller. (Errors that occur on *units* are retrieved using the *UERR* command.) This command returns errors for all clip management commands and other commands that do not pertain to a specific unit.

The response code from this command is 200, and a single response line is returned in the following format:

```
code "description"
```

where

- *code* is the error code.
- *description* is the error description.

FRAT (Set Frame Rate)

To set the frame rate used in translating timecodes for command timing, use the following command:

```
FRAT frame-rate
```

where *frame-rate* is the frame rate in frames per second. Supported values are 24, 25, 29.97, and 30.

Note: The frame rate that is used for command timing defaults to 29.97 if the *FRAT* command is not run.

GTOD (Get System Time)

To get the current VCP-Recorder system time, use the following command:

```
GTOD
```

The response code for this command is 202. A single response line is returned in the following format:

```
timecode UTC-time UST-time
```

where,

- *timecode* is the system time in ISO 8601 format.
- *UTC-time* is the system time in UTC (ISO 8806) format.
- *UST-time* is the Unadjusted System Time. This value is approximately the number of nanoseconds since the server was last booted.

For example:

```
GTOD
202 OK
14:24:01.11 19980105T222400.890307Z 94038459071434
```

MON (Monitor)

To place control connections into event monitoring mode, use the following command:

```
MON [unit-name ... ] [/event-type ... ]
```

where

- *unit-name* is the name of a *unit* for which the control connection should be placed in event monitoring mode. Up to 32 units may be specified.
- *event-type* is a list of one or more event types, each one preceded by a single forward slash. The event types are UADD (unit added), URM (unit removed), UCHG (unit state change), UCTL (unit control change), UERR (unit error), CADD (*clip* added), CRM (*clip* removed). Only the specified types of events are returned. If no *event-type* list is specified, UCHG, URM, and UCTL events are returned. If no units are specified, UADD events are also returned.

The *MON* (Monitor) command places the current control connection into event monitoring mode. In this mode, a single response line is returned whenever a monitored event occurs. If one or more units are specified, unit events for only the specified units are returned; otherwise, unit events for all units are returned.

Event monitoring is terminated by closing the control connection.

The following is the format of the event response line for a UADD:

UADD *owner port mode*

where

- *owner* is the name of the controller that created the unit.
- *port* is the name of the unit's media *port*.
- *mode* is the mode of the media port (IN, OUT, or BOTH).

The following is the format of the event response line for a URM:

URM *unit*

where *unit* is the name of the unit.

The following is the format of the event response line for a UCHG:

UCHG *unit clip status function location speed frame-rate*

where

- *unit* is the name of the unit.
- *clip* is the name of the loaded clip ("*" if no clip is loaded).
- *status* is the status of the current function. PEND is pending, BUSY is waiting to execute, RUN is running, DONE is the completed state, and ERR is the error state. The status can move forward (PEND -> BUSY -> RUN -> DONE -> ERR) but never backward.

Note: In the case of *CUE*, the status reports DONE when the unit is fully cued. However, when a unit is cued, it is sending frames to the video device. Therefore, it is possible that after the status has been reported as DONE, something may happen to create an error. If the *CUE* gets an error after having reported DONE, the subsequent *PLAY* fails because the unit is no longer cued.

- *function* is the current function: IDLE, LOAD, UNLD (Unload), CUE, CUER (Cue for record), PLAY, JOG (Jog), SHTL (Shuttle), REC (Record), PAUS (Pause), STOP, ERR (Error).
- *location* is the current clip location (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).
- *speed* is the current playback speed (1000 = normal 1x speed).
- *frame-rate* is the frame rate of the clip in frames per second. 525-line timing is specified in its approximate form (29.97 frames per second).

The following is the format of the event response line for a UCTL:

```
UCTL unit control-name "control-value"
```

where

- *unit* is the name of the unit.
- *control-name* is the name of the control.
- *control-value* is the new value of the control.

The following is the format of the event response line for a UERR:

```
UERR unit error-code "error-message"
```

where

- *unit* is the name of the unit.
- *error-code* is the code of the error.
- *error-message* is the textual error message.

The following is the format of the event response line for a CADD:

CADD *clip format size resident-size start end in out frame-rate*

where

- *clip* is the name of the clip.
- *format* is the format of the clip. The format is specified using the MediaHub hierarchical syntax.
- *size* is the size of the clip content in bytes.
- *resident-size* is the size of the content that is currently resident on the *clip cache* disk system.
- *start* is the timecode of the first frame of the clip (hh:mm:ss:ff, or hh:mm:ss:ff in drop-frame mode).
- *end* is the timecode of the frame after the last frame of the clip (hh:mm:ss:ff, or hh:mm:ss:ff in drop-frame mode).
- *in* is the current edit in-point of the clip (hh:mm:ss:ff, or hh:mm:ss:ff in drop-frame mode).
- *out* is the current edit out-point of the clip (hh:mm:ss:ff, or hh:mm:ss:ff in drop-frame mode).
- *frame-rate* is the frame rate of the clip in frames per second. 525-line timing is specified in its approximate form (29.97 frames per second).

The following is the format of the event response line for a CRM:

CRM *clip*

where *clip* is the name of the clip.

PLS (List Ports)

To get the list of supported media *ports*, use the following command:

```
PLS
```

This command returns the list of supported media ports. If the command is successful, the response code is 201 and one response line for each port is returned. The format of each line is

```
name mode "description" type
```

where

- *name* is the name of the port.
- *mode* is the port's input/output mode (IN, OUT, or BOTH).
- *description* is a more verbose description of the port.
- *type* is the type of the port, which may be one of the following:
 - VID (video)
 - DECK (deck control)
 - NET (network)
 - DISK (disk storage)¹

For example:

```
PLS
201 OK
vlan_1 BOTH "VLAN Deck Control" DECK
DIVO_0 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_1 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_2 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_3 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_4 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_5 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_6 BOTH "SGI XT-DIVO Digital Video Option" VID
DIVO_7 BOTH "SGI XT-DIVO Digital Video Option" VID
```

¹The network and disk storage ports are not implemented.

SORD (Set Sort Order)

To set the sort order that is used when lists are returned, use the following command:

```
SORD order-type
```

where *order-type* specified the desired sort order. The possible values of *order-type* are

- NAME, which sorts lists by the *clip* name
- TIME, which sorts lists by creation time

The response code for this command is 200.

STLS (List Statistics)

To list statistical information, use the following command:

```
STLS [component-pattern [statistic-pattern]]
```

where

- *component-pattern* is a pattern that describes the names of the components for which you want to see statistics. If this is omitted, the command returns statistics for all components.
- *statistic-pattern* is a pattern that describes the names of the statistical values that you want to see. If this is omitted, the command returns all statistics for the components that are specified.

The asterisk (“*”) can be used when specifying the *component-pattern* or *statistic-pattern* in this command. For example, the following command lists all the statistics for components that contain “DIVO” in their names:

```
STLS *DIVO*
```

If the command is successful, the response code is 201, and a response line is returned for each matching statistical value in the following format:

component-name statistic-name value . . .

where

- *component-name* is the name of the instance of the component that is generating the statistic.
- *statistic-name* is the name of the statistical value.
- *value* is the current value of the integer or floating-point statistic. The value of certain types of statistics (for example, histograms) may include more than one number.

STOD (Set System Time)

To set the VCP-Recorder system time, use the following command:

STOD *time*

where *time* is the time to which the system time is set. The time can be in either the VCP-Recorder timecode format (see “About MVCP Timecodes” on page 57 for information) or the ISO 8601 format (yyyymmddThhmmss.ssssssZ).

STST (Statistics Statistics)

To calculate various statistics, use the following command:

STST [*component-pattern* [*statistic-pattern*]]

where

- *component-pattern* is a pattern that describes the names of the components for which you want to see statistics. If this is omitted, the command returns statistics for all components.
- *statistic-pattern* is a pattern that describes the names of the statistical values that you want to see. If this is omitted, the command returns all statistics for the components that are specified.

The asterisk (“*”) can be used when specifying the *component-pattern* or *statistic-pattern* in this command. For example, the following command calculates all the statistics for components that contain “DIVO” in their names:

```
STST *DIVO*
```

The *STST* (Statistics Statistics) command calculates various statistics over all of the statistical values matching the specified patterns. If the command is successful, the response code is 202 and a single response line is returned in the following format:

```
values samples min max sum mean stddev
```

where

- *values* is the number of statistical values matching the pattern.
- *samples* is the total number of samples collected.
- *min* is the minimum value.
- *max* is the maximum value.
- *sum* is the sum of the values.
- *mean* is the mean of the values.
- *stddev* is the standard deviation of the values.

STZ (Reset Statistics)

To reset the VCP-Recorder statistics, use the following command:

```
STZ [component-pattern [statistic-pattern ]]
```

where

- *component-pattern* is a pattern that describes the names of the components for which you want the statistics reset. If this is omitted, the command resets statistics for all components.
- *statistic-pattern* is a pattern that describes the names of the statistical values that you want reset. If this is omitted, the command resets all statistics for the components that are specified.

The asterisk (“*”) can be used when specifying the *component-pattern* or *statistic-pattern* in this command. For example, the following command resets all the statistics for components that contain “DIVO” in their names:

```
STZ *DIVO*
```

UADD (Add Unit)

To create a new *unit*, use the following command:

```
UADD media-port-name storage-port-name port-access-mode
```

where

- *media-port-name* is the name of the media *port* accessed by the unit.
- *storage-port-name* is the name of the storage port accessed by the unit. If *storage-port-name* is an asterisk (*), the default storage system is used.
- *port-access-mode* specifies how access is shared between multiple units that access the same media port. This parameter takes one of the following values:
 - EXCL, which indicates exclusive access. Only one unit that uses the media port can exist at any given time. If a unit already exists for that port, this command fails.
 - SHAR, which indicates shared access. Multiple units that all access the same port can be created; however, only one unit can actually use the hardware at any given time. (Shared access is available only on media ports that support it in their interface module.)
 - CONC, which indicates concurrent access. Multiple units can use the media port at the same time. This mode is used most commonly on multiplexed networking ports, such as ATM.

If the unit is successfully created, the response code is 202 and a single response line is returned. That line contains the name of the newly created unit.

ULS (List Units)

To get a list of the existing VCP-Recorder *units*, use the following command:

ULS

This command returns a list of the existing VCP-Recorder units along with a snapshot of the state of each unit.

If this command is successful, the response code is 201. One response line for each unit is returned followed by a blank line. The format of each response line is

name owner port mode clip status function location speed frame-rate

where

- *name* is the name of the unit.
- *owner* is the name of the controller that created the unit.
- *port* is the name of the unit's media *port*.
- *mode* is the mode of the media port (IN, OUT, or BOTH).
- *clip* is the name of the loaded *clip* ("*" if no clip is loaded).
- *status* is the status of the current function, which can be one of the following:
 - BUSY, which is the initial state
 - RUN, which is the running state
 - DONE, which is the completed state
 - ERR, which is the error state
- *function* is the current function, which can be one of the following: IDLE, LOAD, UNLD (Unload), CUE, CUER (Cue for record), PLAY, JOG (Jog), SHTL (Shuttle), REC (Record), PAUS (Pause), STOP.
- *location* is the current clip location (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).
- *speed* is the current playback speed (1000 = normal 1x speed).
- *frame-rate* is the frame rate of the clip in frames per second. 525-line timing is specified in its approximate form (29.97 frames per second).

For example:

```

ULS
201 OK
U1 mvcp/originserver mvp BOTH * DONE IDLE * 0 *
U2 mvcp/originserver mvp BOTH * DONE IDLE * 0 *

```

MVCP Unit Commands

Table 4-3 identifies the MVCP *unit* commands that are described in this section. These commands are used with a specific unit.

Table 4-3 MVCP Unit Commands

| Command | Function |
|---------|--|
| CUE | Cue for playback the clip currently loaded in a unit |
| CUER | Cue for recording the clip currently loaded in a unit |
| EDIT | Cue for recording and then begin recording |
| FCLR | Erase the audio and video from specified frames in a clip |
| FF | Fast forward the unit |
| FINS | Move frames within a clip |
| FNEW | Insert blank frames within a clip |
| FOVR | Overwrite frames within a clip |
| FRM | Remove frames from within a clip |
| GET | Get the values of controls |
| GOTO | Jump a unit's transport to a specified location |
| JOG | Jog a unit forward or reverse by the number of frames |
| LIMS | Modify the edit limits being used by the current playback or record function |
| LOAD | Load a clip into a unit |
| PAUS | Pause a unit |
| PLAY | Begin playback on a unit |

Table 4-3 (continued) MVCP Unit Commands

| Command | Function |
|----------------|---|
| REC | Begin recording on a unit or resume normal recording after a PAUS command |
| REVV | Do the equivalent of a cue for playback and then begin playing |
| REW | Fast reverse the unit |
| RHRS | Cue for playback and then begin playing, with the media port switched to end-to-end mode for playback |
| RSUM | Resume playback or recording |
| SET | Set controls for the unit |
| SHTL | Shuttle a unit at a specified speed |
| STOP | Stop playback or recording on a unit |
| UCLS | Close an opened or created unit |
| UERR | Get the code and description of the last error that occurred on a unit |
| UFLS | Flush the command queue for a unit |
| UGIN | Get the owner and port information for a unit |
| UINF | Get the owner and port information for a unit |
| UINT | Interrupt the command currently being processed by a unit |
| ULOC | Return the current transport location |
| UNLD | Unload the clip currently loaded in a unit |
| UOPN | Open an existing unit and make it controllable |
| USTA | Get the status of a unit |
| USYN | Set the default synchronization mode for a unit specified |
| UUWT | Synchronize command execution between two units |
| UWAT | Wait for the completion of the last command that was issued to a unit |

Before you can use these *unit* commands, you must either create the unit or open it, according to the following:

- When you want to use a unit that does not already exist, you create the unit using the global *UADD* command. When you create the unit, you identify the media *port* and you specify the port access mode, which determines how access is shared among multiple units that access the same media port. You can specify exclusive access, shared access, or concurrent access of the media port.
- When you want to share a unit that has already been created, you open the unit using the global *UOPN* command. When you open a unit, it must have previously been created by another control port. Units can be shared with either another MVCP control connection or with another external control processor, such as a station automation system.

Caution: Extreme care must be exercised when sharing control of a unit between two controllers. Interfering with a unit owned by one of the automation protocol processors (for example, a Louth controller) leads to unpredictable behavior.

CUE (Cue for Playback)

To cue the *clip* currently loaded in a *unit* for playback, use the following command:

```
CUE unit-name [ in-point [ out-point [ direction [ passes ] ] ] ]
```

where

- *unit-name* is the name of the unit to which the command applies.
- *in-point* is the in-point, specified as a frame, at which the clip is cued. If *in-point* is specified as "*" or is missing, the edit in-point stored with the clip is used, or, if the edit in-point is not set, the first recorded frame of the clip is used.

This parameter must be specified if *out-point* is specified.

- *out-point* is the edit out-point, specified as a frame. If *out-point* is specified, the clip is cued with the specified out-point, meaning that playback terminates at the specified frame. If *out-point* is specified as "*" or is missing, the edit out-point stored with the clip is used, or, if the edit out-point is not set, no out-point is used and playback continues until the unit is stopped.

- *direction* specifies the playback direction. It can be one of the following:
 - FWD for forward direction
 - BWD for reverse direction
 - F/B for alternating forward and backward direction (that is, swing mode)
 - B/F for alternating backward and forward direction (that is, swing mode)

The default direction is forward (FWD).

Note: If the direction is FWD or F/B, the clip is cued at its in-point. If the direction is BWD or B/F, the clip is cued at its out-point.

- *passes* is an integer that specifies how many passes through the clip are made. The default is one pass. If *passes* is zero, the clip repeats playing indefinitely until the *STOP* command is issued.

Note: A pass in a given direction counts as one. Therefore, a value of at least 2 should be specified for *passes* when *direction* is either F/B or B/F.

If *passes* is specified as -1, the unit is cued in free-range mode. In this mode, *in-point* is used only as the initial location; it does not define the lower limit of playback. If the *vtr.media.clip.limit.start* control is set, it defines the lower limit of playback. If *vtr.media.clip.limit.start* is not set, the beginning of the clip defines the lower limit.

In free-range mode, the upper limit of playback is defined by *out-point*. If *out-point* is not set, the *vtr.media.clip.limit.end* control defines the upper limit of playback. If neither *out-point* nor *vtr.media.clip.limit.end* is set, the end of the clip defines the upper limit.

Note: See the *vcp-recorder-controls(5)* man page for information about the controls.

If either *in-point* or *out-point* is specified, the other may be specified as a duration by adding a “+” prefix character. The following examples all refer to the same edit range:

```
"1:00:01:00 1:00:06:00"  
"1:00:01:00 +5:00"  
"+5:00 1:00:06:00"
```

The following example cues the clip in unit U1 for the alternating forward and backward direction. The clip is cued at 30 seconds and it plays continuously between the 30 second point and the end until the *STOP* command is issued:

```
CUE U1 00:00:30.00 * F/B 0
```

CUER (Cue for Record)

To cue the *clip* currently loaded in a *unit* for recording, use the following command:

```
CUER unit-name [ in-point [ out-point ] ]
```

where

- *unit-name* is the name of the unit to which the command applies.
- *in-point* is the in-point, specified as a frame, at which the clip is cued. If *in-point* is specified as "*" or is missing, the unit cues to the edit in-point stored with the clip.

This parameter must be specified if *out-point* is specified.

- *out-point* is the edit out-point, specified as a frame. If *out-point* is specified, the clip is cued with the specified out-point, meaning that recording will terminate at the specified frame. If *out-point* is specified as "*" or is missing, recording continues until the unit is stopped.

If either *in-point* or *out-point* is specified, the other may be specified as a duration by adding a "+" prefix character. The following examples all refer to the same edit range:

```
"1:00:01:00 1:00:06:00"
```

```
"1:00:01:00 +5:00"
```

```
"+5:00 1:00:06:00"
```

EDIT (Edit)

To cue a *unit* for recording and then begin recording, use the following command:

```
EDIT unit-name [ in-point [ out-point ] ]
```

where

- *unit-name* is the name of the unit to which the command applies.
- *in-point* is the in-point, specified as a frame, at which the *clip* is cued. If *in-point* is specified as "*" or is missing, the edit in-point stored with the clip is used.
This parameter must be specified if *out-point* is specified.
- *out-point* is the out-point, specified as a frame, at which the clip is cued. If *out-point* is specified as "*" or is missing, the edit in-point stored with the clip is used.

The *EDIT* (Edit) command does the equivalent of a *CUER* command followed immediately by a *REC* command.

Note: This command is implemented for only certain devices (for example, VLAN devices).

FCLR (Clear Frames)

To erase the audio and video from specified frames in a *clip*, use the following command:

```
FCLR unit-name in-frame out-frame
```

where

- *unit-name* is the name of the *unit* to which the command applies.
- *in-frame* is the first frame to be erased.
- *out-frame* is the frame after the last frame to be erased.

The *FCLR* command clears (erases) the frames of the clip loaded into the unit, beginning with the frame specified by *in-frame* (inclusive) and ending with the frame specified by *out-frame* (exclusive). The video and audio associated with the cleared frames is removed, but the frames themselves remain. This contrasts with *FRM*, which removes the frames, closing the hole in the clip.

The clip must be loaded for input (IN) or input/output (BOTH).

FF (Fast Forward)

To fast forward a *unit*, use the following command:

```
FF unit-name
```

where *unit-name* is the name of the unit to which the command applies. The unit must have a clip that is either cued or playing. If the unit is idle (that is, a clip is not cued or playing), an error is written to the VCP-Recorder log.

The *FF* (Fast Forward) command fast forwards the unit specified by *unit-name*. The fast forward speed is device-dependent.

FINS (Insert Frames)

To move frames within a *clip*, use the following command:

```
FINS unit-name source-in source-out dest-in
```

where

- *unit-name* is the name of the *unit* to which the command applies.
- *source-in* is the first frame to be moved (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).
- *source-out* is the last frame to be moved (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).
- *dest-in* is the timecode to which the frames are moved (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).

The *FINS* command moves the clip's frames in the range *source-in* to *source-out*, inserting them at the point specified by the timecode *dest-in*. After *FINS*, the duration of the clip is the same.

The clip must be loaded for input (IN) or input/output (BOTH).

FNEW (Insert New Frames)

To insert blank frames within a *clip*, use the following command:

```
FNEW unit-name in-frame out-frame
```

where

- *unit-name* is the name of the *unit* to which the command applies.
- *in-frame* is the location in the clip of the first frame to be inserted (hh:mm:ss:ff, or hh:mm:ss.fff in drop-frame mode).
- *out-frame* is the frame after the last frame that is inserted (hh:mm:ss:ff, or hh:mm:ss.fff in drop-frame mode).

The *FNEW* command opens a hole in the clip and inserts blank frames at the frame specified by *in-frame* (inclusive) up to *out-frame* (exclusive). The number of new frames added is given by *out-frame - in-frame*.

The clip must be loaded for input (IN) or input/output (BOTH).

FOVR (Overwrite Frames)

To overwrite frames within a *clip*, use the following command:

```
FOVR unit-name source-in source-out dest-in
```

where

- *unit-name* is the name of the *unit* to which the command applies.
- *source-in* is the first frame to be moved (hh:mm:ss:ff, or hh:mm:ss.fff in drop-frame mode).
- *source-out* is the frame after the last frame to be moved (hh:mm:ss:ff, or hh:mm:ss.fff in drop-frame mode).
- *dest-in* is the timecode of the destination to be overwritten (hh:mm:ss:ff, or hh:mm:ss.fff in drop-frame mode).

The *FOVR* command moves the frames in the range *source-in* to *source-out*, to the frames starting at the timecode specified by *dest-in*. The frames starting at *dest-in* are overwritten. After the *FOVR* command, the duration of the clip is decreased. The amount depends on whether the source and target frames overlap.

The clip must be loaded for input (IN) or input/output (BOTH).

FRM (Remove Frames)

To remove frames from within a *clip*, use the following command:

```
FRM unit-name in-frame out-frame
```

where

- *unit-name* is the name of the *unit* to which the command applies.
- *in-frame* is the first frame to be removed (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).
- *out-frame* is the frame after the last frame to be removed (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).

The *FRM* command removes the frames of the clip, from the frame specified by *in-frame* (inclusive) to the frame specified by *out-frame* (exclusive). This command removes the frames altogether, moving the frames after the removed frames down in the timeline to close the hole. This contrasts with *FCLR*, which simply erases the video and audio associated with the frames but does not remove the frames themselves.

The clip must be loaded for input (IN) or input/output (BOTH).

GET (Get Control)

To get the values of the controls, use the following command:

```
GET unit-name dev-type control-name-pattern . . .
```

where

- *unit-name* is the name of the *unit* to which the command applies. The *GET* (Get Control) command gets the values of device controls for the unit specified by *unit-name*.
- *dev-type* specifies whether the controls are queried in the media (MED) or storage (STOR) device assigned to the unit. The controls are device-dependent.
- *control-name-pattern* is the name of the control. The asterisk (“*”) can be used when specifying the *control-name-pattern* in this command.

Note: See the `vcp-recorder-controls(5)` man page for information about the controls.

If the command is successful, the response code is 201 and one response line is returned for each matching control. The format of each line is

```
control-name control-value
```

where *control-name* is the name of the control and *control-value* is its value.

The following example gets the values of the device controls that begin with “vtr.media.clip” for unit U1:

```
GET U1 MED vtr.media.clip*
201 OK
vtr.media.clip.limit.end "*"
vtr.media.clip.preset.start "01:00:00:00"
vtr.media.clip.limit.start "*"
vtr.media.clip.format "movie/vframe"
```

GOTO (GO TO)

To jump a *unit*'s transport operation to a specified location, use the following command:

```
GOTO unit-name timecode
```

where

- *unit-name* is the name of the unit to which the command applies.
- *timecode* is the location to which the unit's transport should be jumped (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).

The *GOTO* (Goto) command jumps the unit transport specified by *unit-name* to the location specified by *timecode*. The unit transport continues the function that it was executing starting at the target of the *GOTO* command.

Note: The *GOTO* (Goto) command does not work while the unit is idle.

JOG (Jog)

To jog a *unit* by the number of specified frames, use the following command:

```
JOG unit-name frames
```

where

- *unit-name* is the name of the unit to which the command applies. The unit must be paused before it can be jogged.
- *frames* is the number of frames to jog the unit. If the number is positive, the unit is jogged forward. If the number is negative, the unit is jogged backward.

Note: If a device is *not* frame-accurate, it may report a clip location that is an estimate. After the first time the unit is jogged, however, the location will be frame-accurate.

LIMS (Set Edit Limits)

To modify the edit limits being used by the current playback or record function, use the following command:

```
LIMS unit-name in-point [ out-point [ direction [ passes ] ] ]
```

where

- *unit-name* is the name of the *unit* to which the command applies.
- *in-point* is the in-point, specified as a frame, at which the *clip* is cued.
- *out-point* is the edit out-point, specified as a frame.
- *direction* specifies the playback direction. It can be one of the following:
 - FWD for forward direction
 - BWD for reverse direction
 - F/B for alternating forward and backward direction (that is, swing mode)
 - B/F for alternating backward and forward direction (that is, swing mode)
- *passes* is an integer that specifies how many passes through the clip are made. The default is one pass. If *passes* is zero, the clip repeats playing indefinitely until the *STOP* command is issued.

If *passes* is specified as -1, the unit is cued in free-range mode. In this mode, the unit is cued to the point specified by *in-point*; *out-point* is ignored. In addition, *in-point* is used only as the initial location; it does not define the lower limit of playback.

The *LIMS* (Set Edit Limits) command modifies the edit limits being used by the current playback or record function. The edit limits that can be changed are device-dependent. In general, only the *out-point* can be changed as a way to stop playback or recording at a specific point.

Note: This function is not implemented on Vela ports.

LOAD (Load Clip)

To load a *clip* into a *unit*, use the following command:

```
LOAD unit-name clip-name [load-mode] [CRTE]
```

where

- *unit-name* is the name of the unit to which the command applies.
- *clip-name* is the name of the clip to be loaded.
- *load-mode* indicates whether the clip is being loaded for input (IN), output (OUT), or input and output (BOTH). The default is output (OUT).
- CRTE indicates that the clip should be created if it does not exist.

If the command is successful, the response code is 202. A single response line is returned in the following format:

```
format size resident-size size start end in-point out-point frame-rate
```

where

- *format* is the format of the clip. The format can be one of the following:
 - *movie/vframe/uncompressed* (uncompressed VCP-Recorder variable-frame format movies)
 - *movie/vframe/rice* (compressed VCP-Recorder variable-frame format movies using *Rice* compression)
 - *movie/vframe/jpeg* (motion JPEG)
 - *movie/stream/mpeg/mpeg2/program* (MPEG-2 program stream)
 - *movie/stream/mpeg/mpeg2/transport* (MPEG-2 transport stream)
- *size* is the size of the clip content in bytes.
- *resident-size* the size of the content that is currently resident on the *clip cache* disk system.
- *start* is the timecode of the first frame of the clip (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).
- *end* is the timecode of the frame after the last frame of the clip (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).

- *in-point* is the current edit in-point of the clip (hh:mm:ss:ff, or hh:mm:ss.fff in drop-frame mode).
- *out-point* is the current edit out-point of the clip (hh:mm:ss:ff, or hh:mm:ss.fff in drop-frame mode).
- *frame-rate* is the frame rate of the clip in frames per second. 525-line timing is specified in its approximate form (29.97 frames per second).

PAUS (Pause)

To pause a *unit*, use the following command:

```
PAUS unit-name
```

where *unit-name* is the name of the unit to which the command applies. The unit must be in a playback or record state for this command to take effect.

Note: The *RSUM*, *PLAY*, *JOG*, *SHTL* or *REC* commands are used to resume playback or recording.

If the unit is playing, the video output is a still frame at the pause location.

PLAY (Play Unit)

To begin playback on a *unit*, use the following command:

```
PLAY unit-name speed
```

where

- *unit-name* is the name of the unit to which the command applies. The unit must be cued for playback or in a playback state (that is, paused, shuttle, jog).
- *speed* can be specified as 1000 (1x forward play) or -1000 (1x backward play).

Note: The *PLAY* command is also used to resume normal playback after a *PAUS*, *SHTL*, or *JOG* command.

This command cannot be used to resume playback after a *STOP* command is issued, in which case the unit must be re-cued.

REC (Record)

To begin recording on a *unit* or resume normal recording after a *PAUS* command, use the following command:

```
REC unit-name
```

where *unit-name* is the name of the unit to which the command applies. The unit must be cued for recording or in a recording state (that is, paused).

This command cannot be used to resume recording after a *STOP* command is issued. The unit must be re-cued.

REVV (Review)

To do the equivalent of a *CUE* command followed immediately by a *PLAY* command on a *unit*, use the following command:

```
REVV unit-name [ in-point [ out-point [ direction [ passes ]]]]
```

where

- *unit-name* is the name of the unit to which the command applies.
- *in-point* is the in-point, specified as a frame, at which the *clip* is cued. If *in-point* is specified as "*" or is missing, the edit in-point stored with the clip is used.
This parameter must be specified if *out-point* is specified.
- *out-point* is the out-point, specified as a frame, at which the clip is cued. If *out-point* is specified as "*" or is missing, the edit in-point stored with the clip is used.
- *direction* specifies the playback direction. It can be one of the following:
 - FWD for forward direction
 - BWD for reverse direction
 - F/B for alternating forward and backward direction (that is, swing mode)
 - B/F for alternating backward and forward direction (that is, swing mode)

The default direction is forward (FWD).

Note: If the direction is FWD or F/B, the clip is cued at its in-point. If the direction is BWD or B/F, the clip is cued at its out-point.

- *passes* is an integer that specifies how many passes through the clip are made. The default is one pass. If *passes* is zero, the clip repeats indefinitely until the STOP command is issued.

Note: This command is implemented for only certain devices (for example, VLAN devices).

REW (Fast Reverse)

To fast reverse a *unit*, use the following command:

```
REW unit-name
```

where *unit-name* is the name of the unit to which the command applies. The unit must have a clip that is either cued or playing. If the unit is idle (that is, a clip is not cued or playing), an error is written to the VCP-Recorder log.

The *REW* (Fast Reverse) command fast reverses the unit specified by *unit-name*. The reverse speed is device-dependent.

RHRS (Rehearse)

To do the equivalent of a *CUE* command followed immediately by a *PLAY* command, while switching the media *port* to end-to-end mode during playback, use the following command:

```
RHRS unit-name [ in-point [ out-point [ direction [ passes ]]]]
```

where

- *unit-name* is the name of the *unit* to which the command applies.
- *in-point* is the in-point, specified as a frame, at which the *clip* is cued. If *in-point* is specified as "*" or is missing, the edit in-point stored with the clip is used.
This parameter must be specified if *out-point* is specified.
- *out-point* is the out-point, specified as a frame, at which the clip is cued. If *out-point* is specified as "*" or is missing, the edit in-point stored with the clip is used.

- *direction* specifies the playback direction. It can be one of the following:
 - FWD for forward direction
 - BWD for reverse direction
 - F/B for alternating forward and backward direction (that is, swing mode)
 - B/F for alternating backward and forward direction (that is, swing mode)

The default direction is forward (FWD).

Note: If the direction is FWD or F/B, the clip is cued at its in-point. If the direction is BWD or B/F, the clip is cued at its out-point.

- *passes* is an integer that specifies how many passes through the clip are made. The default is one pass. If *passes* is zero, the clip repeats indefinitely until the *STOP* command is issued.

Note: This command is implemented for only certain devices (for example, VLAN devices).

RSUM (Resume)

To resume the playback or recording function that was paused, use the following command:

```
RSUM unit-name
```

where *unit-name* is the name of the *unit* for which the playback or recording is resumed.

SET (Set Control)

To set the controls for a *unit*, use the following command:

```
SET unit-name dev-type control-name control-value [control-name control-value] . . .
```

where

- *unit-name* is the name of the unit to which the command applies.
- The *dev-type* specifies whether the controls are to be set in the media (MED) or the storage (STOR) device assigned to the unit.
- *control-name* is the name of the control. The controls are device-dependent.
- *control-value* is the value to which the control is set. If the control value contains any spaces or tabs, the value must be enclosed in double quotation marks.

Note: The possible values for *control-name* and *control-value* are defined in the `vcp-recorder-controls(5)` man page.

For each control, a name and a value must be provided. At least one control name/value pair must be specified.

The following example sets the value of a storage control for unit U1 so that frames are automatically cleared from the *clip* after being played:

```
SET U1 STOR vtr.storage.fs.clear_after_play true
200 OK
```

SHTL (Shuttle)

To shuttle a *unit* at a specified speed, use the following command:

```
SHTL unit-name speed
```

where

- *unit-name* is the name of the unit to which the command applies.
- *speed* is the speed at which the unit is shuttled. A speed of 1000 is normal 1x speed. The speed scale is linear, so a speed of 100 is 1/10th normal speed and a speed of 10000 is 10x speed. Positive speeds play forward and negative speeds play backward. (Some devices cannot play backwards.)

Note: The actual shuttle speeds available are device-dependent.

STOP (Stop Unit)

To stop playback or recording on a *unit*, use the following command:

```
STOP unit-name
```

where *unit-name* is the name of the unit to which the command applies. Playback or recording cannot be resumed on this unit until it is re-cued with the *CUE* or *CUER* command, respectively.

UCLS (Unit Close)

To close an opened or created *unit*, use the following command:

```
UCLS unit-name
```

where *unit-name* is the name of the unit to which the command applies. This controller's access is terminated and, if no other controllers have the unit open, the unit is deleted.

If the command is successful, response code 200 is returned.

UERR (Unit Error)

To get the code and description of the last error that occurred on a *unit*, use the following command:

```
UERR unit-name
```

where *unit-name* is the name of the unit to which the command applies.

This command returns the code and description of the last error that occurred on the specified unit.

If the command is successful, the response code 202 is returned followed by a data line in the following format:

```
code "description"
```

where

- *code* is the error code of the last error.
- *description* is a textual description of the error.

UFLS (Flush Unit)

To flush the command queue for a *unit*, use the following command:

```
UFLS unit-name
```

where *unit-name* is the name of the unit whose command queue is flushed.

See “Synchronization of Unit Commands and Responses” on page 62 for a description of the command queue.

UGIN (Get Unit Info)

To get the owner and *port* for a *unit*, use the following command:

```
UGIN unit-name
```

where *unit-name* is the name of the unit for which the owner and port information is returned. The specified unit does not need to be opened by the current control connection. (The *UINF* returns the identical information for an opened unit.)

If the unit exists, the response code is 202 and a single response line is returned with this format:

```
owner port-name port-mode
```

where

- *owner* is the name of the controller that created the unit.
- *port-name* is the name of the media port controlled by the unit.
- *port-mode* is the input/output mode supported by the port. The possible values are IN, OUT, and BOTH.

UINF (Get Unit Info)

To get the owner and *port* information for a *unit*, use the following command:

```
UINF unit-name
```

where *unit-name* is the name of the unit to which the command applies. The unit must have been opened or created by the current controller.

The response code is 202 and a single response line in the following format is returned:

```
owner port-name port-mode
```

where

- *owner* is the name of the controller that created the unit.
- *port-name* is the name of the media port controlled by the unit.
- *port-mode* is the input/output mode supported by the port. The possible values are IN, OUT, and BOTH.

UINT (Unit Interrupt)

To interrupt the command currently being processed by a *unit*, use the following command:

```
UINT unit-name
```

where *unit-name* is the name of unit for which the current command is interrupted. The command is terminated with error EINTR.

ULOC (Unit Location)

To return the current transport location for a *unit*, use the following command:

```
ULOC unit-name
```

where *unit-name* is the name of the unit to which the command applies.

The response code is 202 and a single response line is returned in the following format:

```
clip-loc vitc ltc UTC-time UST-time
```

where

- *clip-loc* is the *clip* location, which roughly corresponds to a CTL (control track) timecode.

Note: The accuracy of the timecode that is returned depends on the capabilities of the device. If a device is *not* frame-accurate, it returns a best-effort estimate of the current location.

- *vitc* is the VITC (vertical interval timecode) of the frame.
- *ltc* is the LTC (longitudinal timecode) of the frame.
- *UTC-time* is the UTC time when the unit was at the specified location. This time can be used to account for application or network latency in time reporting. This time is reported in the ISO 8601-compatible format: `yyyymmddThhmmss.ssssssZ`.
- *UST-time* is the UST (unadjusted system time) when the unit was at the specified location. This time can be used to account for application or network latency in time reporting.

For example:

```
ULOC U2
202 OK
00:00:00.00 00:00:00:00/00:00:00:00 00:00:00:00/00:00:00:00
    19980108T205545.154669Z 174838119321533
```

Note: The *ULOC* command returns a single response line. In this example, the second line is indented to indicate that it is part of the previous line.

UNLD (Unload Clip)

To unload the *clip* currently loaded in a *unit*, use the following command:

```
UNLD unit-name
```

where *unit-name* is the name of the unit to which the command applies.

If this command is successful, the response code is 200.

UOPN (Unit Open)

To open an existing *unit* and make it controllable by this MVCP connection, use the following command:

```
UOPN unit-name
```

where *unit-name* is the name of an existing unit to be opened and made controllable by this control connection.

If the unit is opened successfully, the response code is 202. A single response line is returned containing the name of the newly opened unit.

USTA (Unit Status)

To get the status of a *unit*, use the following command:

```
USTA unit-name
```

where *unit-name* is the name of the unit to which the command applies. The unit must be opened or created by the current controller.

If this command is successful, the response code is 202. A single response line in the following format is returned:

```
clip status function location speed frame-rate
```

where

- *clip* is the name of the loaded *clip* (“*” if no clip is loaded).
- *status* is the status of the current function, which can be one of the following:
 - BUSY indicates the initial state
 - RUN indicates the running state
 - DONE indicates the completed state
 - ERR indicates the error state
- *function* is the current function, which can be one of the following: IDLE, LOAD, UNLD (Unload), CUE, CUER (Cue for record), PLAY, JOG (Jog), SHTL (Shuttle), REC (Record), PAUS (Pause), STOP.
- *location* is the current clip location (hh:mm:ss:ff, or hh:mm:ss.ff in drop-frame mode).
- *speed* is the current playback speed (1000 = normal 1x speed).
- *frame-rate* is the frame rate of the current clip in frames per second. 525-line timing is specified in its approximate form (29.97 frames per second).

USYN (Set Unit Synchronization)

To set the default synchronization, preemption, and queuing modes for a specified *unit*, use the following command:

```
[sync-mode] USYN unit-name
```

where

- *sync-mode* is the default value for the unit. This parameter can have any combination of the synchronization, preemption, and queuing modes described in “Synchronization of Unit Commands and Responses” on page 62.

If this parameter is omitted, the default modes described in “Synchronization of Unit Commands and Responses” on page 62 are used for this unit.

- *unit-name* is the name of the unit to which the command applies.

Once this command has been used to set new default modes for this unit, all subsequent commands for this unit that do not specify a synchronization, preemption, or queuing mode use the default set by the *USYN* command.

The following statement changes the default modes for unit U1 such that a new command will execute after the previous command completes (/SEQ) and the response to the command is delayed until the unit has reached the initialization state on the command (/SYNI):

```
/SEQ /SYNI USYN U1
```

UUWT (Unit Unit Wait)

To synchronize command execution between two *units*, use the following command:

```
UUWT unit-name wait-unit-name sync-mode
```

where *unit-name* is the name of the unit that waits until the unit specified by *wait-unit-name* reaches the execution status specified by *sync-mode*. Both units must be opened by the current control connection.

For the possible values of *sync-mode*, see “Synchronization of Unit Commands and Responses” on page 62.

UWAT (Unit Wait)

To wait for the completion of the last command that was issued to a *unit*, use the following command:

```
UWAT unit-name [sync-mode ]
```

where

- *unit-name* is the name of the unit to which the command applies.
- *sync-mode* is the synchronization mode to use. This parameter can have one of the following values: ASYN, SYNI, SYNRR, and SYNC. (See “Changing the Synchronization Mode” on page 63 for a description of these synchronization modes.)

If this parameter is omitted, the default synchronization mode is used.

This command waits for the last command issued to the unit according to the synchronization mode specified by *sync-mode* or the default synchronization mode, if the *sync-mode* argument is not provided.

Using a MediaHub Archive System

This appendix contains information about MediaHub *assets* that can be brought into VCP-Recorder or that were created by storing a VCP-Recorder *clip* in the *archive system*. The following topics are discussed:

- “About MediaHub Clip Groups” on page 123
- “Bringing MediaHub Clips Into VCP-Recorder” on page 126
- “Finding MediaHub Clips” on page 127
- “Storing VCP-Recorder Clips in MediaHub” on page 128

This chapter includes examples of *MVCP* commands, which are described in detail in Chapter 4, “MVCP Reference.” The examples assume that a telnet connection has been made to the VCP-Recorder server. (For an example of a telnet session, see “Establishing an Interactive MVCP Connection” on page 56.)

This appendix assumes that you are already familiar with the following:

- the VCP-Recorder graphical user interface (GUI), as described in Chapter 2, “Using the VCP-Recorder GUI”
- MediaHub assets, as described in the *MediaHub User’s Guide*

About MediaHub Clip Groups

In order for VCP-Recorder to be able to bring an *asset* into VCP-Recorder, the asset must be a MediaHub group asset that is known as a “clip group.” A clip group is an asset that has a **ClipId** attribute whose value is the name of the clip. The members of the clip group consist of the assets that contain different digital formats of the *clip*.

Note: When VCP-Recorder requests an asset from MediaHub, it requests the asset by sending MediaHub the value of the **ClipId** attribute.

The **DasAsset::Format** attribute of the clip group can be `/group/multiformat` or `/group/format`.

Multiformat clip groups (that is, clip groups whose format is `/group/multiformat`) contain one or more formats of a clip. Each member of a multiformat clip group can be one of the following:

- An atom, if the digital format does not require an index.
- A group whose **DasAsset::Format** is `/group/format`, if the digital format requires an index. Members of each of these group assets are the atom asset that contains the clip's content and the atom asset that contains the index.

If the **DasAsset::Format** attribute of the clip group is `/group/format`, the clip group contains a single format of the clip. The clip group always has a member atom that contains the content. If the digital format requires an index, the clip group also has a member atom that contains the index.

Figure A-1 shows the structure of a multiformat clip group that contains the MPEG-2 and uncompressed *Rice* formats of a movie; each of these formats requires an index. The values of the MediaHub **DasAssetId** and **DasAsset::Format** attributes are shown, along with the value of the **ClipId** attribute. Notice that only the clip group asset has a **ClipId** attribute.

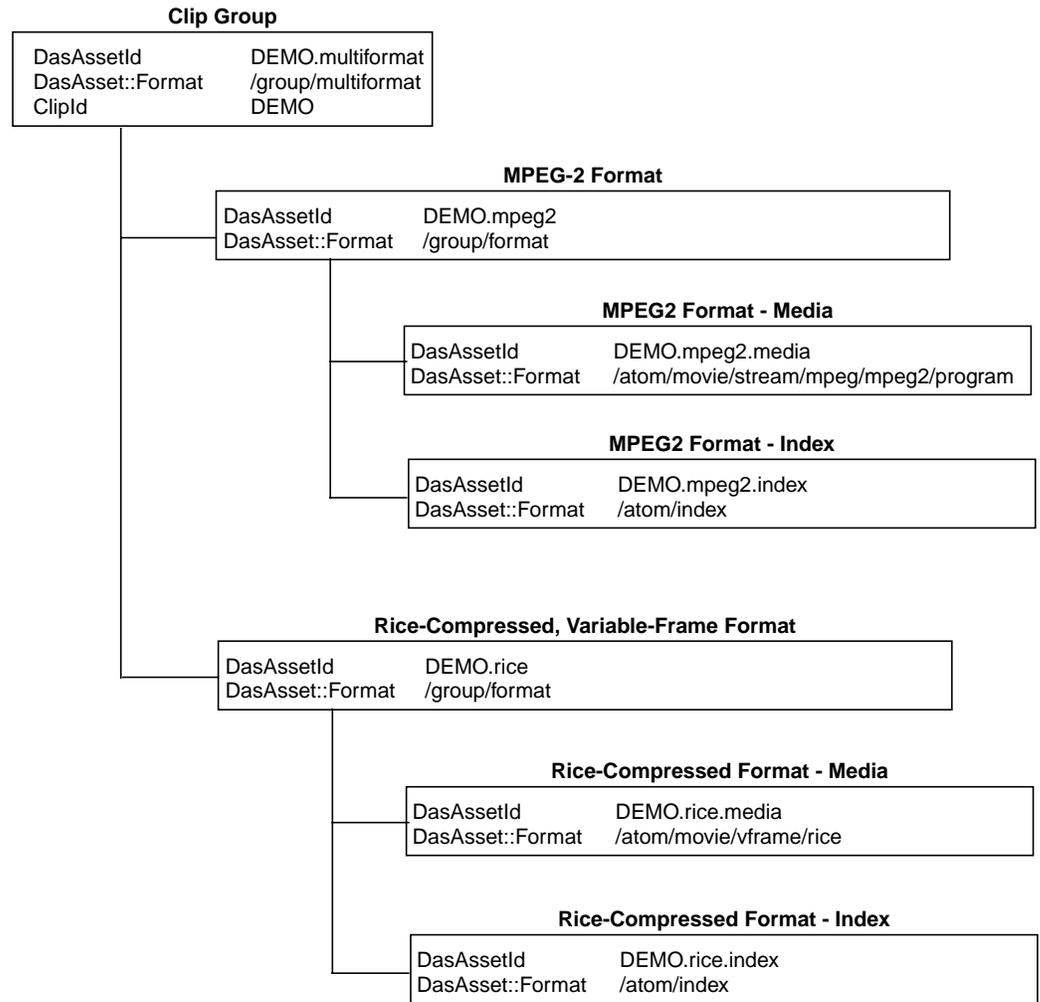


Figure A-1 MediaHub Multiformat Clip Group

Figure A-2 shows the structure of a single-format clip group that contains the MPEG-2 format of a movie. The values of the MediaHub **DasAssetId** and **DasAsset::Format** attributes are shown, along with the value of the **ClipId** attribute. Notice that only the clip group asset has a **ClipId** attribute.

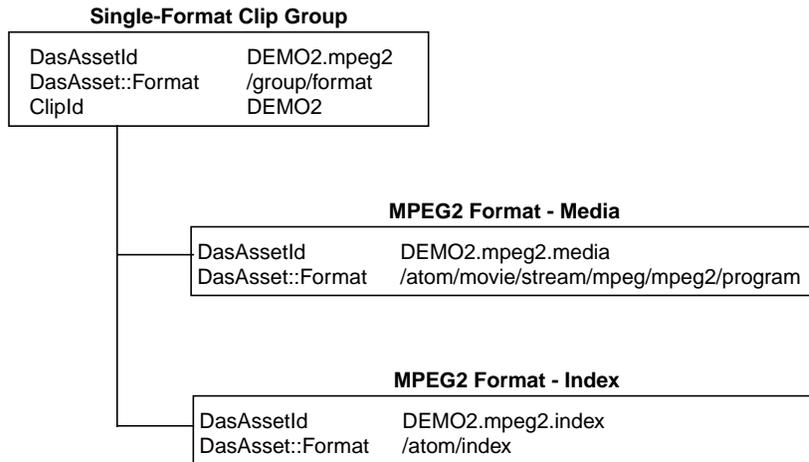


Figure A-2 MediaHub Single-Format Clip Group

Bringing MediaHub Clips Into VCP-Recorder

If a MediaHub clip group adheres to the structure described in “About MediaHub Clip Groups” on page 123, you can use the VCP-Recorder GUI or the MVCP *AGET* (Archive Get) command to request that the *clip* be brought into VCP-Recorder. When you make such a request, the following steps are performed:

1. Through the GUI or the MVCP command, you specify the value of the **ClipId** attribute for the MediaHub clip group. This clip group must contain a format that is compatible with the VCP-Recorder (for example, variable-frame, *Rice*-compressed for a VCP-Recorder that runs on an Origin2000 with DIVO).
2. VCP-Recorder sends a command to the archive system to determine whether an asset with a **ClipId** attribute containing the specified clip name exists. (VCP-Recorder communicates with MediaHub by sending it Asset Transfer Service commands.)
3. If the clip group does not exist, VCP-Recorder writes an error message to its log.

4. If the clip group exists, VCP-Recorder requests information about the formats of the members in the clip group.
 - If a compatible format does not exist, VCP-Recorder writes an error message to its log.
 - If a compatible format exists, VCP-Recorder copies the content and index (if it exists) from the archive system into its *clip cache*.

For example, assume that the clip group shown in Figure A-2 exists in an attached MediaHub archive system and that it contains a suitable format. The following example illustrates how the *AGET* command is used to bring that clip into VCP-Recorder:

```
AGET DEMO2
200 OK
```

Finding MediaHub Clips

You can use VCP-Recorder to determine if a *clip* exists in an attached archive system using either the VCP-Recorder GUI or the MVCP *AFND* (Archive Find) command. The clip group must adhere to the structure described in this appendix and it should contain a suitable media format (for example, variable-frame *Rice*-compressed for a VCP-Recorder that runs on an Origin2000 with DIVO).

The following example uses the *AFND* command to determine that the clip named DEMO2 exists in the MediaHub host named *mhub_host*:

```
AFND DEMO2 mhub_host
201 OK
mediahub mhub_host
```

Storing VCP-Recorder Clips in MediaHub

You can use the VCP-Recorder GUI or the MVCP *APUT* (Archive Put) command to request that a VCP-Recorder *clip* be stored in an attached archive system. When you make such a request, the following steps are performed:

1. Through the GUI or the MVCP command, you specify the name of the clip that you want to store in the archive system.
2. VCP-Recorder sends a command to the archive system to determine whether an asset with a **ClipId** attribute containing the specified clip name exists. (VCP-Recorder communicates with MediaHub by sending it Asset Transfer Service commands.)
 - If the clip group does not exist, VCP-Recorder creates a new multiformat clip group containing the new format¹. It then copies the digital media content and the index file (if it exists) to the archive system.
 - If the clip group exists but it does not contain the targeted media format, VCP-Recorder creates a new single-format group that is added as a member of the MediaHub clip group. It then copies the digital media content and the index file (if it exists) to the archive system.
 - If the clip group exists and it contains the targeted media format, VCP-Recorder requests that the files that contain the content and index for that format be truncated. It then copies the digital media content and the index file (if it exists) to the archive system, replacing the originals.
 - If any MediaHub group assets were created, VCP-Recorder requests that the appropriate member assets be added to those group assets.

For example, the following uses the *APUT* command to store the clip named DEMO3 into the specified archive system:

```
APUT DEMO3 mhub_host
200 OK
```

¹The names of MediaHub assets created by VCP-Recorder are generated automatically by the system.

Glossary

archive system

A repository for storing and distributing digital media data. An *attached* archive system is one that a VCP-Recorder has been configured to use. VCP-Recorder can store *clips* in, and retrieve them from, attached archive systems only.

asset

Unit of storage in an *archive system*. Each asset consists of descriptive information such as the title and duration; digital media data, if the asset has content; and an index, if required by the content format. For example, each movie, commercial, trailer, thumbnail, and so on, stored in an archive system is an asset. Assets may consist of only descriptive information if those assets are used to group other assets.

automation controller

Computers that control broadcast devices. Automation controllers provide features such as playing and recording *clips* according to a predefined schedule, providing statistics about actual events, previewing schedules, controlling broadcast equipment, and so on.

clip

Unit of storage in VCP-Recorder. Each clip consists of descriptive information such as the title and duration; digital media data, if the clip has content; and an index, if required by the content format.

clip cache

XFS filesystems in which VCP-Recorder stores *clips*. The clip cache can reside on a normal XFS filesystem that can be shared with other uses, or a real-time filesystem created on a set of striped disks or RAID units.

cue point

Timecodes that are used to move around within a *clip* and to control the portion of the clip that is played. An in-point, the duration, and an out-point are each specified using the *hh:mm:ss:ff* format, where *hh* is the hours, *mm* is the minutes, *ss* is the seconds, and *ff* is the frame number. In drop-frame mode, the final colon is replaced by a period (*hh:mm:ss.ff*).

For example, if a clip with a cue in-point of 00:00:30.00 is cued for playing in the forward direction, it is cued at 30 seconds.

DIVO

Digital Video Option, available on Origin servers. DIVO provides digital video *ports* through which digital media data is played and recorded by VCP-Recorder.

edit point

Persistent timecode values that are stored with a *clip*. Edit points are used to automatically initialize the *cue points* when the clip is loaded. An in-point, the duration, and an out-point can each be specified using the *hh:mm:ss:ff* format, where *hh* is the hours, *mm* is the minutes, *ss* is the seconds, and *ff* is the frame number. In drop-frame mode, the final colon is replaced by a period (*hh:mm:ss.ff*).

For example, if a clip has an edit in-point of 00:00:30.00, its cue in-point is initialized to 30 seconds when the clip is loaded.

Edit points may also be called “edit marks.”

MVCP

Multiple-Unit Video Computer Protocol. A protocol defined by Silicon Graphics that provides full-featured control of VCP-Recorder through TCP/IP.

port

A point at which an external device connects to VCP-Recorder. VCP-Recorder defines the following types of ports:

- Video ports, which are used by video input and output devices. *DIVO* video ports are named *DIVO_”n”* (for example, *DIVO_2*), the O2 workstation video port is named *mvp*, and Vela decoder ports are named *vela_”n”* (for example, *vela_2*).
- Deck control ports, which are used by VLAN transmitters. Deck control ports are named *vlan_”n”* (for example, *vlan_0*).

The term *media port* refers to any port through which digital media data is passed. That is, the term refers to either a video or a deck control port.

Rice

A lossless bit reduction (compression) encoding scheme that is supported on the Silicon Graphics *DIVO* video option.

unit

A software mechanism that functions like a logical video tape recorder transport that is capable of loading, cueing, playing, and recording digital *clips*. The unit manages a media *port*, which may be shared with another unit.

A unit may also be called a *logical unit*.

VTR

Videotape recorder, which is a device that records and plays videotapes.

Index

A

AFND (Find Clip in Archive) command, 68
AFNG (Find Archive Get) command, 69
AGET (Get From Archive) command, 70
ALSG (List Archive Gets) command, 71
ALSP (List Archive Puts) command, 72
APUT (Put to Archive) command, 73
archive interface modules, 7
archive systems, 1, 7, 49-53, 123-128
 deleting clips, 74
 finding clips, 50, 68, 127
 getting clips, 51, 69, 70, 71, 74, 126
 storing clips, 53, 72, 73, 128
 updating clips, 52
ARM (Delete From Archive) command, 74
ARMG (Cancel Archive Gets) command, 74
automation controllers, 7

B

BYE command, 74

C

CADD (Add Clip) command, 75
CCHP (Change Clip Protection) command, 76
CCP (Copy Clip) command, 77
CCST (Clip Cache Status) command, 77

CEDP (Set Edit Point) command, 78
CGP (Get Clip Protection) command, 78
CIMG (Create Clip Image) command, 79
CINF (Clip Info) command, 80
clip cache, 6, 7, 14
 determining status, 77
 initiate monitoring, 84
Clip Manager, 23, 41-53
clips, 6
 adding, 75
 copying, 77
 copying to file, 79
 creating, 31
 deleting, 47, 85
 erasing frames, 102
 finding, 50, 68, 127
 getting, 51, 69, 70, 71, 74, 126
 getting protections, 78
 inserting blank frames, 104
 inserting frames, 103
 managing, 41-53
 obtaining information, 45, 80, 82, 83, 84
 overwriting frames, 104
 playing, 29
 recording, 29, 31
 removing, 74
 removing frames, 105
 renaming, 46, 85
 setting protections, 48, 76
 sharing media content, 81
 storing, 53, 72, 73, 128
 updating, 52

CLN (Link Clip) command, 81
CLS (List Clips) command, 82
CLSA (List Added Clips) command, 83
CLSR (List Removed Clips) command, 83
CMIN (Clip Monitor Info) command, 84
CMON (Clip Monitor) command, 84
CMV (Rename Clip) command, 85
control interface modules, 4, 7
controls
 getting, 106
 setting, 114
CRM (Delete Clip) command, 85
CRMA (Delete All Clips) command, 85
CUE (Cue for Playback) command, 99
cue buttons, 24, 30
cue points, 23, 32
CUER (Cue for Record) command, 101

D

Deck Control Panel, 33-38

E

EDIT (Edit) command, 102
edit marks, 24, 32
 modifying, 108
 setting, 78
end-to-end mode, 28, 37, 112
ERR (Error) command, 86
errors, determining, 86

F

FCLR (Clear Frames) command, 102
FF (Fast Forward) command, 103
FINS (Insert Frames) command, 103
FNEW (Insert New Frames) command, 104
format interface modules, 4, 8
formats, digital media, 8
FOVR (Overwrite Frames) command, 104
frame rate
 setting, 86
FRAT (Set Frame Rate) command, 86
FRM command, 105

G

GET (Get Control) command, 106
GOTO (GO TO) command, 107
graphical user interface, 15-40, 41-53
GTOD (Get System Time) command, 87

H

hardware overview, 11

J

JOG (Jog) command, 107
JPEG, motion, 8, 31

L

LIMS (Set Edit Limits) command, 108
LOAD (Load Clip) command, 109
Louth automation controllers, 4, 7, 11, 14

M

Media Control Panel, 16-38

- buttons, 26
- local mode, 28
- remote mode, 28
- starting, 16

media device interface modules, 4, 8

MediaHub, 6, 7, 49-53, 123-128

- clip groups, 123

MON (Monitor) command, 87

MPEG, 4, 8

MVCP commands, 55-122

- AFND* (Find Clip in Archive), 68
- AFNG* (Find Archive Get), 69
- AGET* (Get From Archive), 70
- ALSG* (List Archive Gets), 71
- ALSP* (List Archive Puts), 72
- APUT* (Put to Archive), 73
- ARM* (Delete From Archive), 74
- ARMG* (Cancel Archive Gets), 74
- BYE*, 74
- CADD* (Add Clip), 75
- CCHP* (Change Clip Protection), 76
- CCP* (Copy Clip), 77
- CCST* (Clip Cache Status), 77
- CEDP* (Set Edit Point), 78
- CGP* (Get Clip Protection), 78
- changing sort order, 92
- CIMG* (Create Clip Image), 79
- CINF* (Clip Info), 80
- CLN* (Link Clip), 81
- CLS* (List Clips), 82

- CLSA* (List Added Clips), 83
- CLSR* (List Removed Clips), 83
- CMIN* (Clip Monitor Info), 84
- CMON* (Clip Monitor), 84
- CMV* (Rename Clip), 85
- CRM* (Delete Clip), 85
- CRMA* (Delete All Clips), 85
- CUE* (Cue for Playback), 99
- CUER* (Cue for Record), 101
- EDIT* (Edit), 102
- ERR* (Error), 86
- FCLR* (Clear Frames), 102
- FF* (Fast Forward), 103
- FINS* (Insert Frames), 103
- FNEW* (Insert New Frames), 104
- FOVR* (Overwrite Frames), 104
- FRAT* (Set Frame Rate), 86
- FRM* (Remove Frames), 105
- GET* (Get Control), 106
- global commands, 66-97
- GOTO* (GO TO), 107
- GTOD* (Get System Time), 87
- JOG* (Jog), 107
- LIMS* (Set Edit Limits), 108
- LOAD* (Load Clip), 109
- MON* (Monitor), 87
- PAUS* (Pause), 110
- PLAY* (Play Unit), 110
- PLS* (List Ports), 91
- REC* (Record), 111
- response codes, 59
- REVV* (Review), 111
- REW* (Fast Reverse), 112
- RHRS* (Rehearse), 112
- RSUM* (Resume), 113
- sequencing, 62
- SET* (Set Control), 114
- SHTL* (Shuttle), 115
- SORD* (Set Sort Order), 92
- STLS* (List Statistics), 92
- STOD* (Set System Time), 93

STOP (Stop Unit), 115
STST (Statistics Statistics), 93
STZ (Reset Statistics), 94
synchronizing, 62, 121
timecodes, 57
timing, 65
UADD (Add Unit), 95
UCLS (Unit Close), 115
UERR (Unit Error), 116
UFLS (Flush Unit), 116
UGIN (Get Unit Info), 117
UINF (Get Unit Info), 117
UINT (Unit Interrupt), 118
ULOC (Unit Location), 118
ULS (List Units), 96
unit commands, 97-122
UNLD (Unload Clip), 119
UOPN (Unit Open), 119
using, 55-65
USTA (Unit Status), 120
USYN (Set Unit Synchronization), 121
UUWT (Unit Unit Wait), 121
UWAT (Unit Wait), 122
MVCP protocol, 7, 9

P

PAUS (Pause) command, 110
PLAY (Play Unit) command, 110
playing clips
 controlling speed, 29
 direction, 27
PLS (List Ports) command, 91
ports
 listing, 91
 media, 8, 9
 video, 17

R

REC (Record) command, 111
REMU (Review) command, 111
REW (Fast Reverse) command, 112
RHRS (Rehearse) command, 112
Rice compression, 8
RSUM (Resume) command, 113

S

SET (Set Control) command, 114
SHTL (Shuttle) command, 115
shuttle dial, 29
software
 overview, 4
SORD (Set Sort Order) command, 92
statistics, 92, 93, 94
status, 25
STLS (List Statistics) command, 92
STOD (Set System Time) command, 93
STOP (Stop Unit) command, 115
storage device interface modules, 7
STST (Statistics Statistics) command, 93
STZ (Reset Statistics) command, 94
system time
 getting, 87
 setting, 93

T

timecodes, 57

U

UADD (Add Unit) command, 95
UCLS (Unit Close) command, 115
UERR (Unit Error) command, 116
UFLS (Flush Unit) command, 116
UGIN (Get Unit Info) command, 117
UINF (Get Unit Info) command, 117
UINT (Unit Interrupt) command, 118
ULOC (Unit Location) command, 118
ULS (List Units) command, 96
units, 9, 17
 closing, 115
 creating, 99
 cueing for playback, 99
 cueing for recording, 101
 determining transport location, 118
 fast forwarding, 103
 flushing command queue, 116
 getting error codes, 116
 getting information, 117
 interrupting commands, 118
 jogging, 107
 jumping transport, 107
 listing, 96
 loading clips, 109
 modifying edit marks, 108
 monitoring, 87
 opening, 99, 119
 pausing, 110
 playing clips, 111, 112
 recording, 102, 111
 resuming playback, 113
 resuming recording, 111, 113
 rewinding, 112
 setting synchronization mode, 121
 shuttling, 115
 status, 120
 stopping, 115

 synchronizing two units, 121
 unloading clips, 119
 waiting for command completion, 122
Unit Status Monitor, 23, 39-40
UNLD (Unload Clip) command, 119
UOPN (Unit Open) command, 119
USTA (Unit Status) command, 120
USYN (Set Unit Synchronization) command, 121
UUWT (Unit Unit Wait) command, 121
UWAT (Unit Wait) command, 122

V

Vela decoders, 4, 8, 13, 14
video decks, 14
 controlling, 33-38

Tell Us About This Manual

As a user of Silicon Graphics products, you can help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please send the title and part number of the document with your comments. The part number for this document is 007-3620-001.

Thank you!

Three Ways to Reach Us

- To send your comments by **electronic mail**, use either of these addresses:
 - On the Internet: techpubs@sgi.com
 - For UUCP mail (through any backbone site): *[your_site]!sgi!techpubs*
- To **fax** your comments (or annotated copies of manual pages), use this fax number: 650-932-0801
- To send your comments by **traditional mail**, use this address:

Technical Publications
Silicon Graphics, Inc.
2011 North Shoreline Boulevard, M/S 535
Mountain View, California 94043-1389