



# SGI Media Server™ for Broadcast User's Guide

Version VST 2.1.1

007-4403-001

---

## CONTRIBUTORS

Written by Ken Jones

Illustrated by Chris Wengelski and Chrystie Danzer

Production by Karen Jacobson

Engineering contributions by Peter Allen, Achuith Bhandarkar, Sebastian Brings, Philippe Churlet, Sandeep Gupte, Ardene Kannely, Marlon Leon, Jim Montine, Mark Negovan, Matthew Rehrer, Chris Shapter, Richard Stein, and Parkson Wong

Other contributions by Danielle McGurk and Brian Payton

---

## COPYRIGHT

© 2003 Silicon Graphics, Inc. All rights reserved; provided portions may be copyright in third parties, as indicated elsewhere herein. No permission is granted to copy, distribute, or create derivative works from the contents of this electronic documentation in any manner, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

---

## LIMITED RIGHTS LEGEND

The electronic (software) version of this document was developed at private expense; if acquired under an agreement with the USA government or any contractor thereto, it is acquired as "commercial computer software" subject to the provisions of its applicable license agreement, as specified in (a) 48 CFR 12.212 of the FAR; or, if acquired for Department of Defense units, (b) 48 CFR 227-7202 of the DoD FAR Supplement; or sections succeeding thereto. Contractor/manufacturer is Silicon Graphics, Inc., 1600 Amphitheatre Pkwy 2E, Mountain View, CA 94043-1351.

---

## TRADEMARKS AND ATTRIBUTIONS

SGI, the SGI logo, IRIX, Origin, and XFS are registered trademarks and CXFS, Performance Co-Pilot, and SGI Media Server are trademarks of Silicon Graphics, Inc., in the United States and/or other countries worldwide.

Harris Automation and Louth are trademarks of Harris Corporation. Horita is a registered trademark of Horita Co., Inc. Little Red is a trademark of Miranda Technologies, Inc. Panasonic is a registered trademark of Matsushita Electric Industrial Co., Ltd. Sony is a registered trademark of Sony Corporation. UNIX and the X device are registered trademarks of The Open Group in the United States and other countries.

---

## Record of Revision

<b>Version</b>	<b>Description</b>
001	December 2003 Original publication.



# Contents

<b>Figures</b> . . . . .	.xiii
<b>Tables</b> . . . . .	. xv
What This Manual Contains . . . . .	xvii
Who Should Read This Document . . . . .	xviii
Related Documentation . . . . .	xviii
Release Notes . . . . .	.xix
Online VST Man Pages . . . . .	.xix
Program/Utility Files . . . . .	.xix
Command/Control Summaries . . . . .	. xx
Other Publications . . . . .	.xxi
Obtaining Publications . . . . .	.xxi
Conventions . . . . .	.xxii
Reader Comments . . . . .	.xxii
<b>1. Overview of the SGI Media Server.</b> . . . . .	. 1
Functional Overview . . . . .	. 1
Software Overview . . . . .	. 3
Clip Cache . . . . .	. 5
Core Software . . . . .	. 5
Control Interface Module . . . . .	. 5
Storage Device Interface Module . . . . .	. 6
Media Device Interface Module. . . . .	. 6
Compression Formats . . . . .	. 6
Video I/O, Compression, and Formats . . . . .	. 7
Logical Playback and Record Units. . . . .	. 8

Hardware Overview . . . . .	. 12
The Origin Server . . . . .	. 14
Disk Storage . . . . .	. 14
External Devices . . . . .	. 15
Internal Devices . . . . .	. 15
Video Device Control . . . . .	. 16
Supported SMPTE Standards . . . . .	. 16
<b>2. System Setup . . . . .</b>	<b>. 17</b>
Power Requirements . . . . .	. 18
Rack Space Requirements . . . . .	. 19
Sample Space Requirements for Model 320 or Model 325 . . . . .	. 19
Sample Space Requirements for Model 380 or Model 385 . . . . .	. 20
PCI Configuration Requirements . . . . .	. 20
PCI Bus Requirements . . . . .	. 20
General Guidelines for Card Placement . . . . .	. 21
Recommended Card Layouts . . . . .	. 22
A Basic Two-Channel System (Model 320 and Model 325) . . . . .	. 22
A Maximum Eight-Channel System (Model 380 and Model 385) . . . . .	. 23
Video/Audio Connections . . . . .	. 25
PCI-VIDAUD-MSB-B Video/Audio Cards . . . . .	. 26
The MSB-REARPANEL-B Rear Panel . . . . .	. 26
Video Connections . . . . .	. 28
SDI Video Connectors . . . . .	. 28
Genlock/Composite Connectors . . . . .	. 29
Audio Connections . . . . .	. 29
AES/EBU Audio In Connectors . . . . .	. 29
AES/EBU Audio Out Connectors . . . . .	. 30
ANALOG AUDIO Connectors . . . . .	. 30
<b>3. System Configuration . . . . .</b>	<b>. 31</b>
Configuration Files . . . . .	. 31

Configuring Device Interfaces . . . . .	34
Configuring Audio and Video Devices in Your System . . . . .	34
Setting Device Parameters . . . . .	34
Configuring System Defaults. . . . .	35
Configuring Video Compression . . . . .	36
Configuring Audio Support . . . . .	39
Input . . . . .	39
Output . . . . .	41
Channel Map/Source Control . . . . .	42
Channel Map/Channel Pairs Control . . . . .	42
Channel Map Examples . . . . .	43
Identifying Audio Ports . . . . .	44
Disabling Audio . . . . .	44
Configuring the SGI Media Server for Control by Remote Devices . . . . .	45
Control Port Configuration Line . . . . .	45
<i>protocol</i> . . . . .	46
<i>type</i> . . . . .	46
<i>port</i> . . . . .	46
<i>speed</i> . . . . .	47
<i>parity</i> . . . . .	47
<i>signalport</i> . . . . .	47
<i>rate</i> . . . . .	47
<i>latency</i> . . . . .	48
<i>clips=path/*</i> . . . . .	48
Signal Configuration Line . . . . .	48
Example control-in.conf File . . . . .	49
Configuring the SGI Media Server to Control Other Devices . . . . .	50
<i>protocol</i> . . . . .	51
<i>connection</i> . . . . .	51
<i>serial_port</i> . . . . .	51
<i>speed</i> . . . . .	51
<i>parity</i> . . . . .	51
Configuring the VTR Daemon (vtrd) . . . . .	52

Mapping Physical Ports to Logical Port Names. . . . .	. 52
Configuring for 625/50 Systems . . . . .	. 53
Tuning the SGI Media Server (Setting the Maximum DMA Size) . . . . .	. 53
<b>4. Running the SGI Media Server . . . . .</b>	<b>. 55</b>
Starting and Stopping VST . . . . .	. 55
Controlling the Autostart of VST . . . . .	. 56
Starting VST Manually . . . . .	. 56
Stopping VST . . . . .	. 56
Setting Startup Options . . . . .	. 57
Checking VST Status . . . . .	. 57
Logging Events . . . . .	. 57
VST Log Message Structure . . . . .	. 58
Configuring Logging . . . . .	. 59
Managing Log Rollover . . . . .	. 60
Monitoring the System. . . . .	. 61
vtrstat . . . . .	. 61
Standard IRIX Tools . . . . .	. 62
<b>5. Adding and Removing Clips. . . . .</b>	<b>. 63</b>
Overview of Adding Clips Procedure . . . . .	. 64
Media Types and Formats. . . . .	. 65
Media Types . . . . .	. 65
Media Formats . . . . .	. 65
Adding Clips to the SGI Media Server . . . . .	. 66
Transferring Clips to the SGI Media Server . . . . .	. 66
Proxy Transfers . . . . .	. 67
MPEG-2 Index File Transfer . . . . .	. 69
Clip Alignment . . . . .	. 69
Degree of Alignment . . . . .	. 70
Minor Alignment . . . . .	. 70
Major Alignment . . . . .	. 70
Notifying the SGI Media Server . . . . .	. 71
Transferring a Clip Segment . . . . .	. 71



	Overriding Clip Segment Transfer . . . . .	72
	Removing Clips . . . . .	72
	MXF Record and Playout . . . . .	72
<b>6.</b>	<b>Virtual Clips . . . . .</b>	<b>75</b>
	Virtual Clip Command Overview . . . . .	76
	File Operations on Virtual Clips . . . . .	77
	Creating and Opening Virtual Clips . . . . .	77
	Building a Non-Virtual Clip . . . . .	77
	Saving, Closing, and Removing Virtual Clips . . . . .	78
	Working with Segments . . . . .	78
	Adding Segments to Vclips . . . . .	78
	Listing Segments in a Vclip . . . . .	81
	Removing Segments in Vclips . . . . .	82
	Working with Frames . . . . .	83
	Inserting Empty Frames into a Vclip . . . . .	83
<b>7.</b>	<b>Configuring and Using External Devices . . . . .</b>	<b>85</b>
	Configuring and Using the Sony (P2) Protocol. . . . .	86
	Configuring Sony Protocol Control Ports . . . . .	87
	Changing the Clip Loaded in a Sony Controlled Logical Unit . . . . .	89
	Configuring the SGI Media Server to Control a VTR . . . . .	89
	Configuring the SGI Media Server for Control by VDCP. . . . .	91
	VDCP Commands . . . . .	92
	Using the Harris Automation ADC100 Automation Controller with the SGI Media Server . . . . .	94
	Using House Timecode Readers with the SGI Media Server . . . . .	96
	RS-232 and RS-422 Cabling . . . . .	97
<b>8.</b>	<b>Troubleshooting . . . . .</b>	<b>101</b>
	625/50 Clips Do Not Play. . . . .	101
	Adding a Clip Takes Excessively Long . . . . .	101
	Crash: Semaphore Limit Exceeded . . . . .	102
	Configuring to Boot From a Backup Plex . . . . .	102
	SGI Media Server Crash Files . . . . .	103

<b>9. Introduction to MVCP</b>	105
Manual Access to Video Server Technology	106
Creating and Deleting a Unit	107
Deleting a Unit	107
Multiple Connections to a Unit	107
Loading, Creating, and Unloading a Clip	108
Finding the Name of a Clip	108
Setting Edit Points	108
Cueing Decks to Play or Record	109
Sequencing Commands	109
Playing a Prerecorded Clip	109
Setting and Listing Configuration Values	110
Listing Video and Deck Control Ports	111
Assigning Audio Inputs	111
Configuring Video Recording Compression	112
Recording a Clip	112
Editing Clips	113
Editing at the Clip Level	113
Editing at the Frame Level	114
Displaying Your Logo	114
Playing Clips from a Playlist	115
Monitoring Unit State	116
Monitoring the System	117
Status	117
Monitoring	117
Statistics	118
STLS—List Statistics	119
STST—Statistics on Statistics	119
Error Reporting	120
MVCP Command Syntax Errors	120
Controller Errors	120
Unit Errors	120

<b>10.</b>	<b>Clip Mirror Configuration</b>	. . . . .	.121
	Clip Mirror Service Overview	. . . . .	.122
	Automatic Clip Cache Transfers	. . . . .	.122
	Primary and Mirror Platform Failures	. . . . .	.123
	Controlling Clip Mirror Operation	. . . . .	.123
	Parameter <code>vtr.clipmirror.primary_server.hostname</code>	. . . . .	.124
	Parameter <code>vtr.clipmirror.local_server.hostname</code>	. . . . .	.125
	Parameter <code>vtr.clipmirror.max_threads</code>	. . . . .	.125
	Parameter <code>vtr.clipmirror.reconnect_interval</code>	. . . . .	.125
	Configuring Primary and Mirror Platform Filesystems	. . . . .	.126
	Redesignating Servers.	. . . . .	.128
	Bandwidth Management	. . . . .	.128
<b>A.</b>	<b>The SGI Media Server Control GUIs</b>	. . . . .	.131
	Running VST Graphical Tools from a Workstation	. . . . .	.132
	The Media Control Panel	. . . . .	.132
	Starting the Media Control Panel	. . . . .	.133
	Determining Available Ports	. . . . .	.135
	Using Telnet to Determine Available Ports	. . . . .	.136
	Using <code>vtrstat</code> to Determine Units in Use	. . . . .	.136
	Using Telnet to Determine Units in Use	. . . . .	.136
	Sample Startups of the Media Control Panel	. . . . .	.137
	Operating the Media Control Panel.	. . . . .	.138
	Playing or Recording an Existing Clip	. . . . .	.145
	Creating a New Clip	. . . . .	.148
	Changing Cue Points and Edit Points	. . . . .	.149
	Controlling a Video Deck	. . . . .	.150
	Accessing the <b>Deck Control</b> Window	. . . . .	.151
	Using the Deck Control Window	. . . . .	.152
	Recording From the Deck to a Clip	. . . . .	.155
	Recording From a Clip to the Deck	. . . . .	.156
	The <b>Unit Status Monitor</b>	. . . . .	.157

The Clip Manager . . . . .	158
Starting the Clip Manager . . . . .	158
Clip Manager Menus . . . . .	161
Obtaining Information About a Clip . . . . .	162
Renaming a Clip . . . . .	163
Deleting a Clip . . . . .	163
Setting the Protections for a Clip. . . . .	164
<b>Glossary.</b> . . . . .	167
<b>Index.</b> . . . . .	173

# Figures

<b>Figure 1-1</b>	Functional Overview . . . . .	2
<b>Figure 1-2</b>	Video Server Technology Software Components . . . . .	4
<b>Figure 1-3</b>	One Logical Unit with One Control Connection . . . . .	9
<b>Figure 1-4</b>	One Logical Unit with Two Control Connections . . . . .	10
<b>Figure 1-5</b>	Two Logical Units with One Control Connection . . . . .	11
<b>Figure 1-6</b>	SGI Media Server Hardware Components . . . . .	13
<b>Figure 2-1</b>	MSB-REARPANEL-B Left-Half Front View . . . . .	27
<b>Figure 2-2</b>	MSB-REARPANEL-B Left-Half Rear View . . . . .	27
<b>Figure 6-1</b>	Virtual Clips. . . . .	75
<b>Figure 6-2</b>	CUPS Arguments . . . . .	80
<b>Figure 6-3</b>	Copying Versus Linking; Where Segment Data is Stored . . . . .	81
<b>Figure 6-4</b>	Clearing Versus Removing Segments . . . . .	83
<b>Figure 7-1</b>	Origin 300 DB-9 Pinouts for RS-232 and RS-422 Protocols. . . . .	98
<b>Figure A-1</b>	<b>Media Control Panel</b> . . . . .	.135
<b>Figure A-2</b>	<b>Media Control Panel</b> With a Clip Loaded . . . . .	.138
<b>Figure A-3</b>	Loading a Clip into the Logical VTR. . . . .	.146
<b>Figure A-4</b>	<b>Create Clip</b> Window . . . . .	.148
<b>Figure A-5</b>	<b>Deck Control</b> Window . . . . .	.151
<b>Figure A-6</b>	mclistat Output. . . . .	.157
<b>Figure A-7</b>	<b>Clip Manager</b> Window . . . . .	.160
<b>Figure A-8</b>	The Clip Information Window . . . . .	.162
<b>Figure A-9</b>	<b>Rename Clip</b> Window. . . . .	.163
<b>Figure A-10</b>	<b>Delete Clip</b> Window . . . . .	.164
<b>Figure A-11</b>	Set-Protections Window . . . . .	.165



## Tables

<b>Table 1-1</b>	Support for Compression Types . . . . .	7
<b>Table 1-2</b>	Hardware Components of the SGI Media Server . . . . .	12
<b>Table 1-3</b>	SMPTE Standards Supported by the SGI Media Server 300 Systems. . . . .	16
<b>Table 2-1</b>	Power Requirements for the SGI Media Server Base Unit . . . . .	18
<b>Table 2-2</b>	Power Requirements for the SGI Media Server PCI Expansion Module Power Bay . . . . .	18
<b>Table 2-3</b>	Power and Size Specifications for the SGI Media Server External RAID Storage System . . . . .	19
<b>Table 2-4</b>	Two-Channel Model 320/325 Space Requirements . . . . .	19
<b>Table 2-5</b>	Eight-Channel Model 380/385 Space Requirements . . . . .	20
<b>Table 2-6</b>	Supported Bus Speeds for Optimal Add-On Cards . . . . .	21
<b>Table 2-7</b>	Origin 300 Internal PCI Slot Layout . . . . .	22
<b>Table 2-8</b>	Origin 350 Internal PCI Slot Layout . . . . .	23
<b>Table 2-9</b>	PE Brick Layout for Origin 300 Systems . . . . .	24
<b>Table 2-10</b>	PE Brick Layout for Origin 350 Systems . . . . .	25
<b>Table 2-11</b>	SDI Video Connectors . . . . .	28
<b>Table 2-12</b>	GENLOCK/COMPOSITE Connectors . . . . .	29
<b>Table 2-13</b>	AES/EBU Audio Input Connectors . . . . .	29
<b>Table 2-14</b>	AES/EBU Audio Output Connectors . . . . .	30
<b>Table 2-15</b>	Analog Audio Connector . . . . .	30
<b>Table 3-1</b>	Configuration Files. . . . .	32
<b>Table 3-2</b>	Subdirectories of /usr/vtr . . . . .	33
<b>Table 3-3</b>	Supported GOP Structure and Period . . . . .	37
<b>Table 3-4</b>	Example GOP Size, Period, and Structure . . . . .	38
<b>Table 3-5</b>	Source Selections for Record Input Pairs . . . . .	40

<b>Table 3-6</b>	Audio Output Mappings . . . . .	. 41
<b>Table 4-1</b>	Logging Levels . . . . .	. 58
<b>Table 4-2</b>	rotatelogs Command Options . . . . .	. 60
<b>Table 6-1</b>	MVCP Virtual Clip Commands . . . . .	. 76
<b>Table 7-1</b>	VDCP Commands Supported by the SGI Media Server . . . . .	. 92
<b>Table 7-2</b>	Harris Automation Device Parameters . . . . .	. 94
<b>Table 7-3</b>	Origin 300 DB-9 Pin Assignments . . . . .	. 98
<b>Table 7-4</b>	Configuration for a Mini-Din to DB-9 Adapter . . . . .	. 99
<b>Table 7-5</b>	Adapter for Harris Automation Controller (DB-9 to RJ12/16) . . . . .	. 99
<b>Table 7-6</b>	Pin Assignments for an RS-422 Serial Port to an RJ-45 Cable Connector . . . . .	100
<b>Table 9-1</b>	Time Reported by the Unit and the Actual Video Frame . . . . .	117
<b>Table 10-1</b>	Clip Mirror Control Parameters . . . . .	124
<b>Table 10-2</b>	FTP Control Parameters . . . . .	129
<b>Table A-1</b>	Media Control Panel Options . . . . .	133
<b>Table A-2</b>	Log Severity Levels . . . . .	134
<b>Table A-3</b>	Log Severity Levels . . . . .	158



---

## About This Manual

The SGI Media Server for broadcast is a completely integrated solution for serving video, audio, and data that is based on open architecture, data networking, and storage technology. The core of the multichannel SGI Media Server is the SGI Video Server Technology (VST) software. The current product supports the DVCPRO 25, DVCPRO 50, and MPEG-2 formats.

This manual describes how to use the SGI Media Server to play and record digital media. Also described in this manual are the graphical user interfaces (GUIs) that are used to manually control the SGI Media Server, and the Multiport Video Computer Protocol (MVCP), which is a command-line control protocol supported by VST.

## What This Manual Contains

The following material is covered in this manual:

- Chapter 1, “Overview of the SGI Media Server,” provides a functional overview of the SGI Media Server and then describes its software and hardware.
- Chapter 2, “System Setup,” describes the general setup requirements of your SGI Media Server.
- Chapter 3, “System Configuration,” describes the various configuration requirements and the mechanisms for configuring and controlling device-independent and device-specific features.
- Chapter 4, “Running the SGI Media Server,” describes starting, monitoring, and stopping the Video Server Technology (VST) software.
- Chapter 5, “Adding and Removing Clips,” describes how you add and remove clips from the SGI Media Server.
- Chapter 6, “Virtual Clips,” describes virtual clips and how you manipulate them on the SGI Media Server.

- Chapter 7, “Configuring and Using External Devices,” describes how to handle the external devices connected to the SGI Media Server.
- Chapter 8, “Troubleshooting,” describes some likely problem scenarios you might experience while operating your SGI Media Server and what to do.
- Chapter 9, “Introduction to MVCP,” describes some common tasks you routinely perform using MVCP commands.
- Chapter 10, “Clip Mirror Configuration,” describes how you can set up one or more servers for clip mirroring with clip caches that mirror the clip cache on a designated primary server.
- Appendix A, “The SGI Media Server Control GUIs,” describes the graphical user interfaces (GUIs) that you can use to monitor and control system activities (such as playing and recording clips) in non-production environments.

The glossary provides definitions of key words used in this document.

## Who Should Read This Document

This document is written for SGI Media Server users, 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

In addition to the information in this document, you should refer to the documentation described in the following sections:

- “Release Notes” on page xix for information on how to read the online release notes for VST
- “Online VST Man Pages” on page xix for online help for VST commands and utilities
- “Other Publications” on page xxi for a list of related hardware and software publications

## Release Notes

The VST release notes provide information about the release that is not included in this document. Included are known issues with installation and setup, product issues, and documentation errata.

You can view the VST release notes by using one of the following two utilities:

`relnotes`      Text-based viewer for online release notes

`grelnotes`     Graphics viewer for online release notes

To list the VST release notes, enter one of the following commands at a shell prompt on the SGI Media Server:

```
relnotes vst_eoe
```

or

```
grelnotes vst_eoe
```

## Online VST Man Pages

Online information provided with the SGI Media Server includes man pages for specific commands and utilities. Refer to these man pages for information about a command. The man page categories and titles are presented in this section.

The man pages are formatted in the standard UNIX “man page” style. You can list the man pages by entering the following command at the command prompt on the SGI Media Server:

```
% versions long vst_eoe | grep man
```

## Program/Utility Files

The following are man pages for program and utility files:

<b>Man Page</b>	<b>Description</b>
<code>vst(1)</code>	Introduces the VST (Video Server Technology) software.
<code>vtrstart(1)</code>	Describes VST startup.

<code>vtrstop(1)</code>	Describes VST shutdown.
<code>vtrstat(1)</code>	Describes how to get VST status.
<code>vtrclip(1)</code>	Describes clip management.
<code>vvtr(1)</code>	Describes the VST server.
<code>vtrd(1)</code>	Describes the VST daemon.
<code>vtrvfutil(1)</code>	Describes the VST vframe clip utility.
<code>vtrdircopy(1)</code>	Describes the copy-directory daemon for VST.
<code>vtrftp(1)</code>	Describes the command-line FTP client for VST.
<code>vtrftpd(1)</code>	Describes the real-time-enabled FTP daemon for VST.
<code>vtrmpegutil(1)</code>	Describes the MPEG-2 parsing utility for VST.
<code>vtrsyncinfo(1)</code>	Describes the VST video sync information tool.
<code>rotatelogs(1)</code>	Describes the rotating of VST log files.

### Command/Control Summaries

The following man pages describe the configuration and control protocol of VST:

<b>Man Page</b>	<b>Description</b>
<code>mvcp(5)</code>	Describes the Multiport Video Computer Protocol (MVCP) commands.
<code>vst-controls(5)</code>	Describes the VST device-independent and device-specific controls.

## Other Publications

Refer to the following publications for additional information:

- *IRIX Admin: Software Installation and Licensing* for information about installing software that runs under IRIX, the SGI implementation of the UNIX operating system
- *IRIX Admin: System Configuration and Operation* for information about IRIX system administration tasks
- *IRIX Admin: Disks and Filesystems* for information about general filesystem concepts and system administration procedures for SCSI disks, XFS and EFS filesystems, logical volumes, and guaranteed rate I/O
- *SGI Origin 300 User's Guide* for information about the server platform that is used for the SGI Media Server Model 320 and Model 380
- *SGI Origin 350 Server System User's Guide* for information about the server platform that is used for the SGI Media Server Model 325 and Model 385
- *PCI Expansion Module User's Guide* for information about the PCI expansion module that is used in the SGI Media Server Model 380

## Obtaining Publications

You can obtain SGI documentation in the following ways:

- See the SGI Technical Publications Library at <http://docs.sgi.com>. Various formats are available. This library contains the most recent and most comprehensive set of online books, release notes, man pages, and other information.
- If it is installed on your SGI system, you can use InfoSearch, an online tool that provides a more limited set of online books, release notes, and man pages. With an IRIX system, select **Help** from the Toolchest, and then select **InfoSearch**. Or you can type `infosearch` on a command line.
- You can also view release notes by typing either `grelnotes` or `relnotes` on a command line.
- You can also view man pages by typing `man <title>` on a command line.

## Conventions

The following conventions are used throughout this publication:

Convention	Meaning
<code>command</code>	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
<i>variable</i>	Italic typeface denotes variable entries and words or concepts being defined.
<b>user input</b>	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. (Output is shown in nonbold, fixed-space font.)
[]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element can be repeated.
manpage( <i>x</i> )	Man page section identifiers appear in parentheses after man page names.
<b>GUI element</b>	This font denotes the names of graphical user interface (GUI) elements such as windows, screens, dialog boxes, menus, toolbars, icons, buttons, boxes, fields, and lists.

## Reader Comments

If you have comments about the technical accuracy, content, or organization of this document, contact SGI. Be sure to include the title and document number of the manual with your comments. (Online, the document number is located in the front matter of the manual. In printed manuals, the document number is located at the bottom of each page.)

You can contact SGI in any of the following ways:

- Send e-mail to the following address:  
techpubs@sgi.com

- Use the Feedback option on the Technical Publications Library webpage:  
<http://docs.sgi.com>
- Contact your customer service representative and ask that an incident be filed in the SGI incident tracking system.
- Send mail to the following address:  
Technical Publications  
SGI  
1500 Crittenden Lane, M/S 535  
Mountain View, CA 94043-1351

SGI values your comments and will respond to them promptly.





## Overview of the SGI Media Server

This chapter describes the SGI Media Server, which provides real-time, frame-accurate recording and playback of broadcast-quality digital media data. The SGI Media Server manages video as data, distributing the files over the existing LAN/WAN infrastructure—whether 100 Base-TX, Gigabit Ethernet, or enhanced video networks—within a facility or between facilities.

The following topics are discussed in this chapter:

- “Functional Overview” on page 1
- “Software Overview” on page 3
- “Hardware Overview” on page 12
- “Supported SMPTE Standards” on page 16

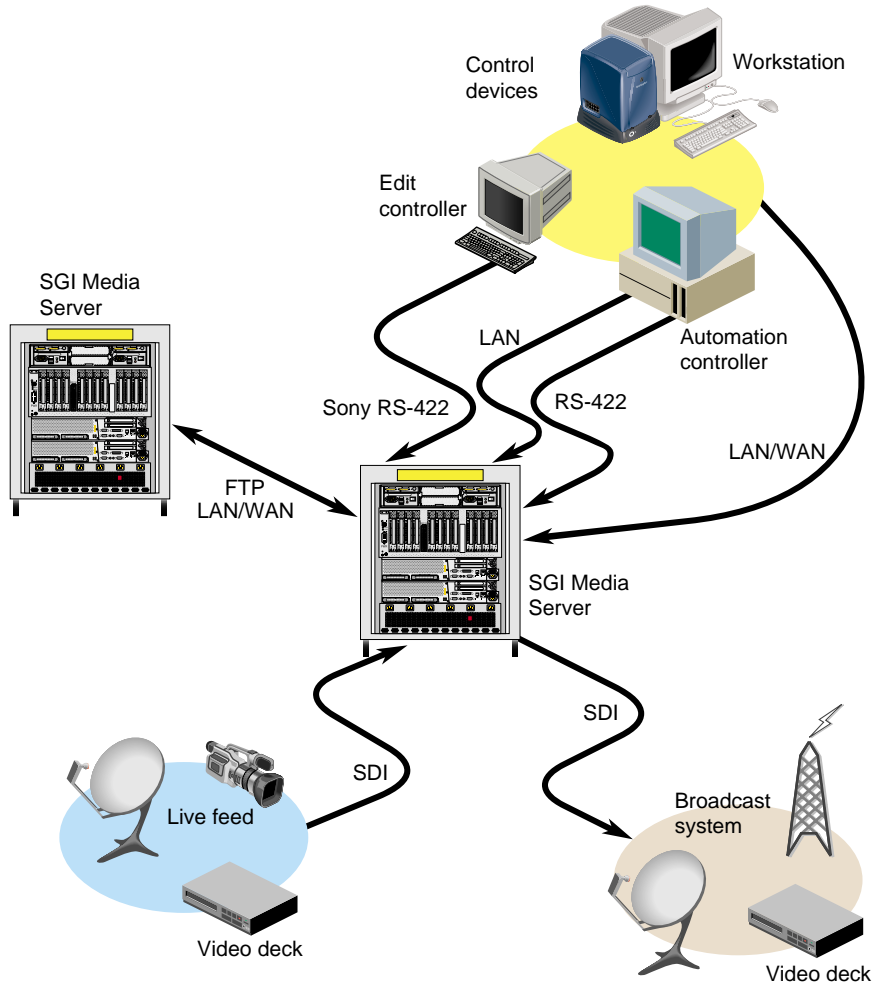
### Functional Overview

Digital media data is brought into the SGI Media Server by recording it from a live feed or a videotape deck or copying it from a file. The data can then be played out to a broadcast system, a video port, or a video tape recorder (VTR).

The SGI Media Server can be controlled using a Multiport Video Computer Protocol (MVCP) application or through the use of a broadcast system automation controller, which can control video servers using serial control protocols—such as the Video Disk Communications Protocol (VDCP) or an edit controller that can control a VTR using the Sony RS-422 VTR protocol.

For non-production, developmental activities, you can use the SGI Media Server control graphical user interface (GUI). The GUI consists of screens that are used to record and play digital data, determine status information, and manage digital media data stored on the SGI Media Server.

This functionality is shown in Figure 1-1.



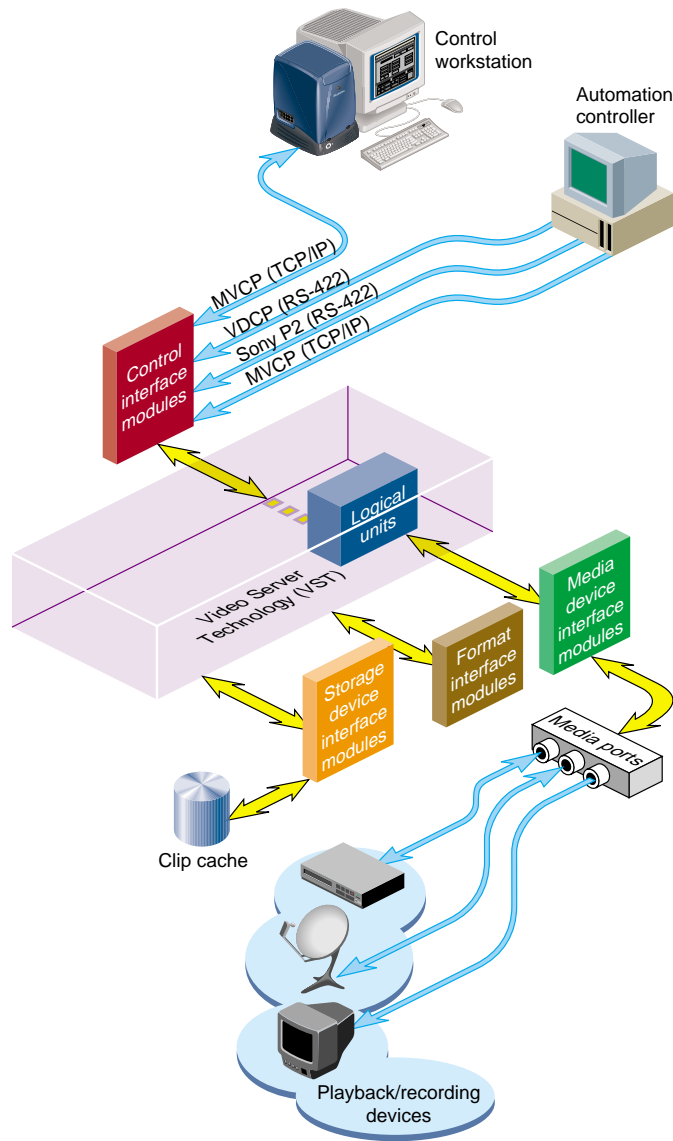
**Figure 1-1** Functional Overview

## Software Overview

The SGI Media Server software provides scalability and maximum flexibility. The software includes the following:

- SGI Video Server Technology (VST), the core software of the SGI Media Server. VST provides real-time, frame-accurate recording and playback of broadcast-quality digital media.
- Control interface modules, which provide device-dependent code. For example, there is a control interface module that contains the code that is specific to VDCP controllers.
- Media device interface modules, which contain format-dependent code that provides access to the *ports* over which media is played and recorded.
- Format interface modules, which provide handlers for accessing specific digital media storage formats. For example, there are format interface modules for MPEG-2, DVCPRO 25, and DVCPRO 50 compression.

Figure 1-2 shows the primary software components in VST. The remainder of this section describes the software components shown in Figure 1-2.



**Figure 1-2** Video Server Technology Software Components

## Clip Cache

Digital media data that the SGI Media Server processes for playout and recording is stored in one or more *clip caches*. Each audio/video segment that is stored in a clip cache is called a *clip*.

Clips can be added to the cache by doing the following:

- Using the SGI Media Server to record the clip
- Generating the clips elsewhere and adding them to the cache by data transfer

## Core Software

The VST core software provides the basic functionality for playback and recording of media. It uses the IRIX operating system as well as portions of the SGI Digital Media Libraries.

The core software provides the following basic functionality:

- **Clip cache management**  
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**  
Links one or more external control protocol modules (for example, VDCP) to the internal processing logic.
- **Configuration management**  
Automatically configures the VST software according to the hardware capabilities of the system on which it runs.

VST provides a core library that supports external interface modules that contain the code specific to a given external entity. When the SGI Media Server is started, the VST software loads and initializes all external interface modules it locates so that the modules can be used.

## Control Interface Module

Control interface modules allow various *automation controllers* and digital media applications to control the use of the SGI Media Server. These modules translate to and from external control protocols.

The following control interface modules are provided:

- The Video Disk Communications Protocol (VDCP). This control protocol provides control of VST over RS-422. The VDCP interface module supports back-to-back play and record (subject to restrictions imposed by the video I/O port capabilities).
- The Sony RS-422 VTR (also called, 9-pin or P2) protocol. VST supports this protocol through a full-featured VTR deck-emulation mode that includes frame-accurate insert editing and a variable-speed shuttle.
- Multiport Video Computer Protocol (MVCP) defined by SGI. This control protocol provides full-featured control of VST through TCP/IP. This control interface module supports archive management, multiple-unit control, and event monitoring and provides access to advanced features of SGI 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.

## 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.

VST has media device interface modules for the following:

- PCI-VIDAUD-MSB-B is the DVCPRO 25, DVCPRO 50, IMX, and MPEG-2 multiformat video and audio codec for PCI backplanes.
- Diaquest (limited support): direct Sony RS-422 control of videotape decks.

## Compression Formats

The MPEG-2, DVCPRO 25, and DVCPRO 50 compression algorithms can be used to provide a compressed video stream, which can be stored in a file. This file format is useful for transporting video across data networks. In addition to traditional TCP/IP data networks, this data can also be transmitted across coaxial video networks as data to

and from VTRs, nonlinear editors, and servers. The following section explains how to display this compressed data stream during transfer.

## Video I/O, Compression, and Formats

The SGI Media Server uses Serial Digital Interface e (SDI) protocols for primary video I/O operations, which are SMPTE-259M compliant. Additionally, the server employs composite analog video for Genlock input and output monitoring. The SGI Media Server is configurable for either MPEG-2, DVCPRO 25, or DVCPRO 50 compression types. For MPEG-2, you can select I-Frame, long GOP, and optionally IMX/constant-byte GOP (CBG). Supported file formats include DIF DVCPRO 25 (SMPTE-314M), DIF DVCPRO 50 (SMPTE-314M), MXF D10/IMX (SMPTE-356M), MXF OP-1a (MPEG-2) (SMPTE-378M), MXF OP-1a (DVCPRO 25/50 - playback only)(SMPTE-378M) and SGI MPEG-2. See “Supported SMPTE Standards” on page 16 for a list of supported standards.

Table 1-1 summarizes the support for the various compression types.

**Table 1-1** Support for Compression Types

Compression Type	Play	Record
MXF MPEG-2	x	x
MPEG-2 I-frame VBR (variable byte rate)	x	x <sup>1</sup>
MPEG-2 IMX 50	x	x <sup>1</sup>
MPEG-2 long GOP <sup>2</sup>	x	x
DIF DVCPRO 25	x	x
DIF DVCPRO 50	x	x
MXF DVCPRO 25	x	
MXF DVCPRO 50	x	

1. For information on limitations of long GOP support, see the VST release notes.
2. Only one of the compression schemes IMX and I-frame VBR is supported at the same time on IMX-record-enabled boards. For playout, both IMX and I-frame VBR are supported simultaneously.

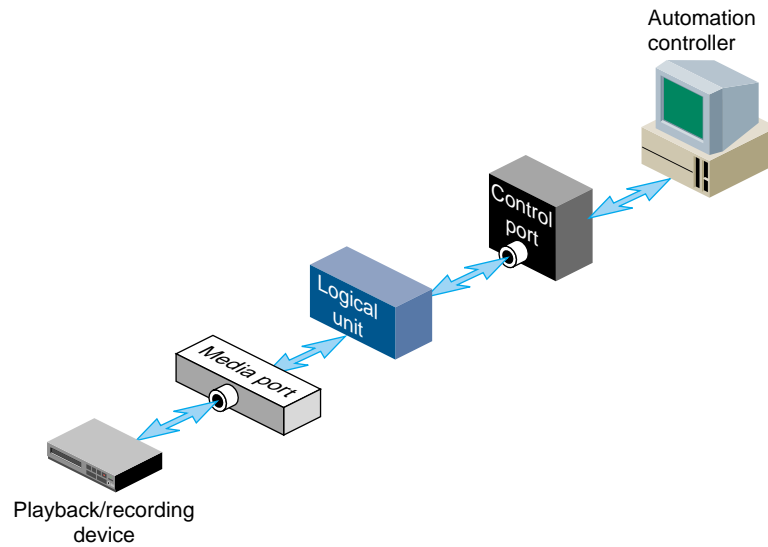
## Logical Playback and Record Units

Logical *units* enable media *ports* to play and record clips. Each SGI Media Server unit can be thought of as a logical VTR transport that is capable of loading, cueing, playing, and recording clips using a specific media port.

Logical units are created automatically by VST when the SGI Media Server control GUI or an automation controller is used. When MVCP is used, a command requests that a unit be added or that a unit created by another control connection be used.

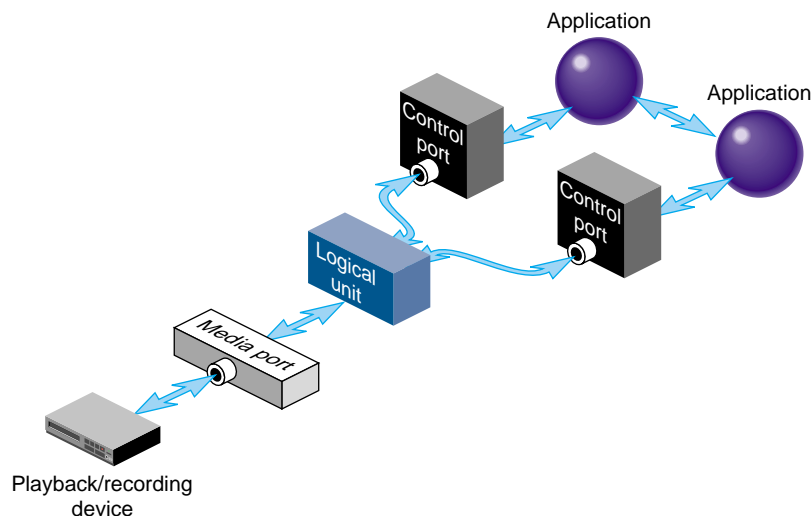


There is normally a one-to-one relationship between a control connection to the SGI Media Server and a VST logical unit and between a logical unit and a media port. This is shown in Figure 1-3.



**Figure 1-3** 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 1-4.



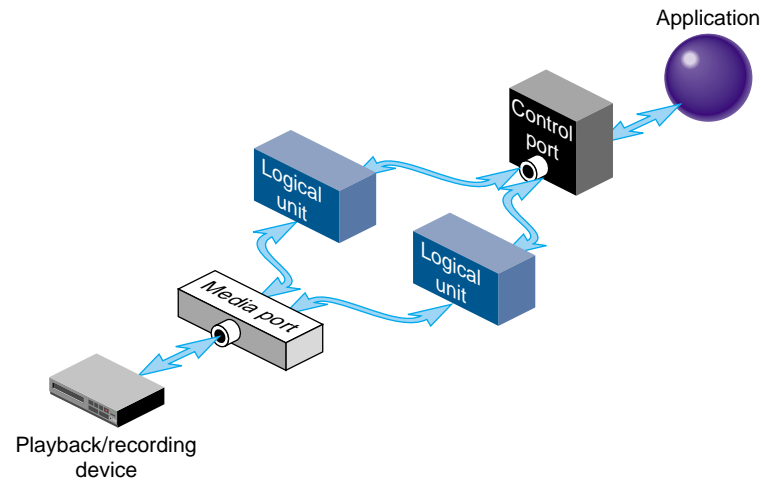
**Figure 1-4** One Logical Unit with Two Control Connections



**Caution:** You can have multiple control connections to control the same unit, but you should do so only if you use extreme caution. When multiple controllers control a single unit, commands coming from the individual control connections can interfere with one another to cause unpredictable and often undesirable behavior. For more information on command sequencing and synchronization, see the `mvcp (5)` man page.

---

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 when allowed by the media format. This example is shown in Figure 1-5.



**Figure 1-5** 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.

## Hardware Overview

This book describes the SGI Media Server Models 320, 325, 380, and 385. Their hardware configuration consists of the components shown in Table 1-2.

**Table 1-2** Hardware Components of the SGI Media Server

Component	Description
Base Unit	Origin 300 or Origin 350 server
Storage	External Fibre Channel or UltraSCSI arrays SGI TP900 SCSI storage systems SGI TP9100 storage systems
Internal Devices	Audio/Video I/O: PCI-VIDAUD-MSB-B or PCI-VIDAUD-IMX cards with video codecs and audio I/O (AES/EBU, embedded, or analog)
Data networking	100 Base-TX 1000 Base-SX
External Devices	Devices using SGI MVCP with TCP/IP  Devices using the Sony/P2 RS-422 protocol—for example, a Panasonic AG-A850 editing controller, a Buf VTC-4000 multi-VTR controller (with or without Bufnet/RS-422 adapter) or a Buf RM-4000 remote processor  Devices using the VDCP RS-422 protocol—for example, Harris Automation controllers

---

**Note:** PCI-VIDAUD-IMX cards are functionally equivalent to PCI-VIDAUD-MSB-B cards, except that the PCI-VIDAUD-IMX cards have unique firmware to enable IMX ingest processing—constant-byte GOP (CBG). Subsequent references to PCI-VIDAUD-MSB-B cards in this manual also apply to PCI-VIDAUD-IMX cards unless otherwise noted.

---

A typical hardware configuration is shown in Figure 1-6.

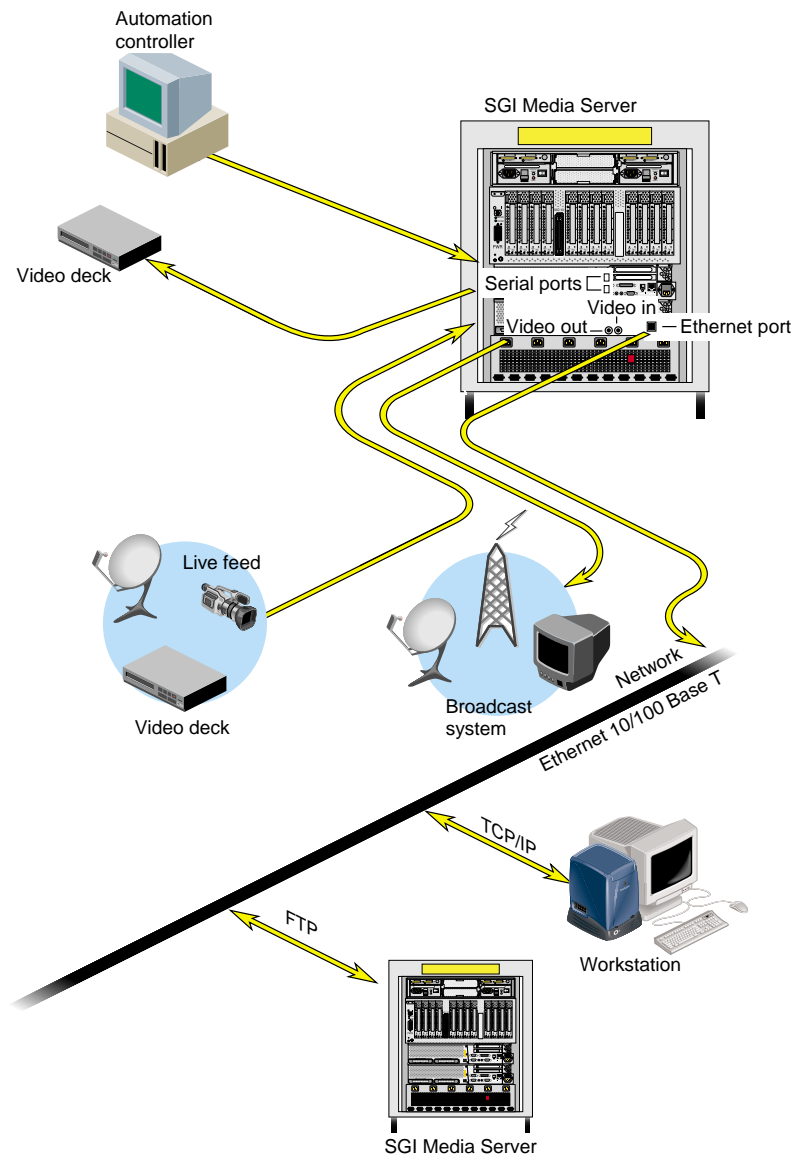


Figure 1-6 SGI Media Server Hardware Components

## The Origin Server

The Origin 300 or Origin 350 server provides massive processing, storage, and throughput capabilities to satisfy large production requirements. They are built from a scalable node architecture, enabling small configurations that can be incrementally upgraded to the larger configurations. Each Origin server can be configured as a single module or as multiple modules with a single system image.

The Origin server provides the following:

- Up to two PCI-VIDAUD-MSB-B boards that support the following:
  - MPEG-2 (I-frame, IMX, and long GOP), DVCPRO 25 and DVCPRO 50 compression
  - The native encoding file formats SGI MPEG-2 for MPEG-2 compression and DIF for DVCPRO 25 and DVCPRO 50
  - The MXF file format for MPEG-2 compression for record and playout
  - For playback only, the MXF file format for DVCPRO 25 and DVCPRO 50
- A PE brick expansion box, which allows up to six additional PCI-VIDAUD-MSB-B boards other add-on components

## Disk Storage

The SGI Media Server disk storage holds the XFS real-time filesystems that contain the video content stored in the clip cache. The descriptive information about clips—for example, clip names, duration, edit points, and so on—are stored on non-real-time filesystems.

The 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.

Two different types of disk storage are available:

- SCSI 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, but it does provide higher bandwidth than XFS on a single disk drive.
- RAID storage, which provides high-availability, redundant digital storage.

## External Devices

The SGI Media Server supports the use of the following external devices:

- Control devices using MVCP over TCP/IP
- Broadcast system automation controllers that use, among others, VDCP. Automation controllers are connected to the server through an RS-422 serial connection.
- Edit controllers, including the following:
  - Panasonic AG-A850 editing controller (Sony/P2 RS-422 protocol).
  - Buf VTC-4000 multi-VTR controller (with or without Bufnet/RS-422 adapter) and Buf RM-4000 remote processor (Sony/P2 RS-422 protocol).
- Timecode readers, including the following:
  - Miranda TCP-101i Time Code Processor and Little Red LTC Reader
  - Horita Master Timecode Generator and PR-232 SMPTE Time Code Translator

For more information about installing these devices for use with the SGI Media Server, see Chapter 7, “Configuring and Using External Devices.”

For a complete list of external devices that interface with the SGI Media Server, see the release notes.

## Internal Devices

The SGI Media Server supports the use of the following internal devices:

- Audio/Video I/O: PCI-VIDAUD-MSB-B cards, which support the MPEG-2, DVCPRO 25, and DVCPRO 50 video codecs and embedded, AES/EBU, and analog audio.

---

**Note:** Only IMX-upgraded cards support D-10/IMX encoding.

---

- Data networking: 100 Base-TX, 1000 Base-SX.

## Video Device Control

The SGI Media Server can also control remote video devices directly using Sony RS-422 deck control software provided by Diaquest (limited support).

## Supported SMPTE Standards

The SGI Media Server provides integrated support for the following SMPTE standards. The SGI Media Server is designed to interact with signal formats and media file formats that follow the SMPTE standards and constraints.

**Table 1-3** SMPTE Standards Supported by the SGI Media Server 300 Systems

Standard	Description
SMPTE 259M-1997 - for Television	10-Bit 4:2:2 Component and 4fsc Composite Digital Signals - Serial Digital Interface (SDI)
SMPTE 266M-2002 - for Television	4:2:2 Digital Components Systems - Digital Vertical Interval Time Code
SMPTE 272M-1994 - for Television	Formatting AES/EBU Audio and Auxiliary Data into Digital Vertical Ancillary Data Space
SMPTE 276M-1995 - for Television	Transmission of AES/EBU Digital Audio Signals Over Coaxial Cable
SMPTE 314M-1999 - Television	Data Structure for DV-Based Audio, Data and Compressed Video - 25 and 50MB/s
SMPTE 356M-2001 - Television	Type D-10/IMX Stream Specifications - MPEG-2 4:2:2:P @ ML for 525/60 and 625/50
SMPTE 378M	Material Exchange Format (MXF) Operational Pattern 1a (OP-1a) specification



## System Setup

This chapter describes the power, rack space, PCI, and video/audio connections required to install and set up an SGI Media Server. The following sections detail these requirements:

This chapter describes setting up your system in the following sections:

- “Power Requirements” on page 18
- “Rack Space Requirements” on page 19
- “PCI Configuration Requirements” on page 20
- “Video/Audio Connections” on page 25

## Power Requirements

Power requirements for the Origin 300 or Origin 350 base platform supporting the SGI Media Server Model 320, 325, 380, and 385 servers are shown in Table 2-1.

**Table 2-1** Power Requirements for the SGI Media Server Base Unit

Specification	Value
Input volts	110 or 220 volts, 20 amp universal auto-switching power supply
Power bay output	640 watts DC (3.3/5/12v)
Cooling	N+1 redundant cooling

Power bay requirements for the PCI expansion module are shown in Table 2-2.

**Table 2-2** Power Requirements for the SGI Media Server PCI Expansion Module Power Bay

Specification	Value
Input volts	110 or 220 volts. The power bays are not universal and do not autoswitch. The default power bay has two offline switcher (OLS) power supplies. They should be in slots 4 and 5 in the power bay. One power bay can drive six I/O bricks with power. Up to six OLSs can be installed in one power bay. A power bay supplies AC voltage and monitors and controls the peripheral I/O brick power units.
Power output	Each distributed power supply inputs single-phase AC power and can output a maximum of 950 watts at 48 VDC. The outputs are bussed together to provide a maximum of 3,800 watts of available peak power in an N+1 redundant configuration. This means there is one additional 950 watt power supply for reserve. The reserve power supply is ready-standby and will power the system in case of failure.
Cooling	N+1 redundant cooling units. A minimum of two power supplies must be present.

Table 2-3 shows the power and size specifications for external RAID storage systems.

**Table 2-3** Power and Size Specifications for the SGI Media Server External RAID Storage System

Specification	Value
Input Volts	200 to 240 VAC, 50 to 60 Hz, 3 A
Dimensions	17.7 cm x 4.46 cm x 58.6 cm (6.95" x 1.75" x 23") for height, width, and depth, respectively
Weight	Approximately 44 kgs (97 pounds)

## Rack Space Requirements

A 2-channel SGI Media Server requires three standard rack units and an 8-channel SGI Media Server requires seven rack units. The SGI Media Server occupies a standard 19-inch rack. In addition, each external SGI TP900 disk array requires 2 rack units, and each SGI TP9100 disk array requires 3 rack units each and resides in a standard 19-inch rack.

The following sections show the space requirements for sample configurations for the SGI Media Server.

### Sample Space Requirements for Model 320 or Model 325

Table 2-4 shows the rack space requirements for a two-channel Model 320 or Model 325 with TP900 storage.

**Table 2-4** Two-Channel Model 320/325 Space Requirements

Component	Rack Space (U)
Origin 300 or Origin 350 (2-CPU) with 2 PCI-VIDAUD-MSB-B cards	2
1 MSB-REARPANEL-B panel	1
TP900 1x8 disk array	2

## Sample Space Requirements for Model 380 or Model 385

Table 2-5 shows the rack space requirements for an eight-channel Model 380 or Model 385 with TP9100 storage.

**Table 2-5** Eight-Channel Model 380/385 Space Requirements

Component	Rack Space (U)
Origin 300 or Origin 350 (4-CPU) with 2 PCI-VIDAUD-MSB-B cards	2
PCI expansion module with 6 PCI-VIDAUD-MSB-B cards	4
4 MSB-REARPANEL-B panels	4
1 TP9100-2G FC-AL disk array	3
Power bay	3

## PCI Configuration Requirements

This section presents general requirements for the PCI bus and sample configurations for the SGI Media Server.

### PCI Bus Requirements

For optimal performance the PCI video/audio cards (PCI-VIDAUD-MSB-B cards), storage, and networking cards should not be mixed with cards of different types while sharing the same bus. Table 2-6 outlines supported bus speeds for each type of add-on card.

**Table 2-6** Supported Bus Speeds for Optimal Add-On Cards

Add-On Cards	33Mhz Bus Speed	66Mhz Bus Speed
Gigabit Ethernet, Tigon-2 (eg <i>n</i> )	X	X
Gigabit Ethernet, Tigon-3 (tg <i>n</i> )	X	X
Fibre Channel host bus adapter (HBA)	X	X
PCI-VIAUD-MSB-B card	X	
Serial I/O card	X	

Cards that support only 33Mhz bus speed will cause the PCI bus to lock to 33Mhz. This can degrade performance of networking and storage cards that support 66Mhz bus speeds or higher.

## General Guidelines for Card Placement

In the section “Recommended Card Layouts” on page 22, specific card layouts are recommended. Some optional/alternative configuration recommendations are also made. The following is a list of general guidelines for card placement on an SGI Media Server:

- Do not use cards of different types on the same bus. However, in fully configured systems this may not be possible.
- PCI-VIAUD-MSB-B boards are always to be put in the specific slots shown in section “Recommended Card Layouts” on page 22.
- Never mix a PCI-VIAUD-MSB-B card with networking or Fibre Channel/SCSI disk HBA cards.
- PCI-VIAUD-MSB-B cards may share the same bus with serial boards, if necessary.
- Serial boards may share the same bus with Gigabit Ethernet cards. However, there will be some loss of networking performance.
- Keep disk and networking cards on separate buses. However, they can be put on the same bus but there may be some overall degradation in both disk and network performance.

---

**Note:** SGI does not support configurations that share serial boards with Fibre Channel HBAs on the same bus.

---

## Recommended Card Layouts

The tables in this section recommend card layouts for the SGI Media Server infrastructure. They represent only standard minimum and maximum configuration options.

The following two conventions are used in the tables:

- Optional slot layout possibilities are in parentheses.
- The descriptors “best choice” and “last choice” are used for serial cards to indicate card placement priority, where “last choice” indicates that you only use this bus/slot if no other slots are available.

### A Basic Two-Channel System (Model 320 and Model 325)

Table 2-7 shows the layout of the two internal PCI slots available on the Origin 300. The Origin 300 also provides a console port (serial I/O) and two additional integrated serial ports that can be used for VDCP control of two PCI-VIDAUD-MSB-B boards. The integrated UltraSCSI interface can be used with the TP900 (if configured). A primary network interface (TX-100) is also provided.

**Table 2-7** Origin 300 Internal PCI Slot Layout

Bus	Slot	Recommended Card Layout for Internal Slots Module (001c02)	Xtalk	Slot
1	1	PCI-VIDAUD-MSB-B card1	9	1
1	2	PCI-VIDAUD-MSB-B card2	9	2

The Origin 350 provides the same integrated interfaces as the Origin 300 with the following exceptions:

- The integrated network adaptor supports (TX-1000).
- There is one additional PCI slot that can be configured as shown in Table 2-8.

**Table 2-8** Origin 350 Internal PCI Slot Layout

Bus	Slot	Recommended Card Layout for Internal Slots Module (001c01)	Xtalk	Slot
2	1	PCI-VIDAUD-MSB-B card1	15	1
2	2	PCI-VIDAUD-MSB-B card2	15	2
1	1	(Gigabit Ethernet)(33Mhz-only cards—last choice)	15	1

See the introductory text for section “Recommended Card Layouts” on page 22 for conventions used in the table.

### A Maximum Eight-Channel System (Model 380 and Model 385)

Table 2-9 and Table 2-10 show the recommended layout for the PE brick on SGI Media Server configurations for both Origin 300 and Origin 350 base units.

The bus and slot information is printed on back of the PE brick just above each slot. The xtalk/slot numbers are available by using the command `vt rhwinfo`. Each PCI-VIDAUD-MSB-B board will report its xtalk/slot number and the table in this section can be used to find the appropriate bus/slot number. This applies for all other cards as well.

See the introductory text for section “Recommended Card Layouts” on page 22 for conventions used in the following tables.

Table 2-9 pertains to Origin 300 systems.

**Table 2-9** PE Brick Layout for Origin 300 Systems

<b>Bus</b>	<b>Slot</b>	<b>Recommended Card Layout for PE Brick Module (001p02)</b>	<b>Xtalk</b>	<b>Slot</b>
1	1	Gigabit Ethernet	9	1
1	2	(Gigabit Ethernet)(Serial—last choice)	9	2
2	1	Fibre Channel	8	1
2	2	(Fibre Channel)	8	2
3	1	PCI-VIDAUD-MSB-B card3	15	1
3	2	PCI-VIDAUD-MSB-B card4	15	2
4	1	PCI-VIDAUD-MSB-B card5	14	1
4	2	PCI-VIDAUD-MSB-B card6	14	2
5	1	PCI-VIDAUD-MSB-B card7 (Serial)	12	1
5	2	PCI-VIDAUD-MSB-B card8 (Serial)	12	2
6	1	Serial—best choice	13	1
6	2	Serial—best choice	13	2



Table 2-10 pertains to Origin 350 systems.

**Table 2-10** PE Brick Layout for Origin 350 Systems

<b>Bus</b>	<b>Slot</b>	<b>Recommended Card Layout for PE Brick Module (001p02)</b>	<b>Xtalk</b>	<b>Slot</b>
1	1	(Gigabit Ethernet)	9	1
1	2	(Serial--last choice)	9	2
2	1	Fibre Channel	8	1
2	2	(Fibre Channel)	8	2
3	1	PCI-VIDAUD-MSB-B card3	15	1
3	2	PCI-VIDAUD-MSB-B card4	15	2
4	1	PCI-VIDAUD-MSB-B card5	14	1
4	2	PCI-VIDAUD-MSB-B card6	14	2
5	1	PCI-VIDAUD-MSB-B card7 (Serial)	12	1
5	2	PCI-VIDAUD-MSB-B card8 (Serial)	12	2
6	1	Serial—best choice	13	1
6	2	Serial—best choice	13	2

## Video/Audio Connections

The SGI Media Server can contain up to eight channels of video I/O, employing the SGI PCI-VIDAUD-MSB-B card. Connections to the card are made through the MSB-REARPANEL-B component. This section provides connection and configuration information for the rear panel and the audio/video card.

## PCI-VIDAUD-MSB-B Video/Audio Cards

PCI-VIDAUD-MSB-B cards are half-size, 64-bit, 33MHz, 3.3V PCI cards. The cards, along with the VST software, provide encoding and decoding of the following video formats:

- SGI MPEG-2
- DIF (DVCPRO 25 and DVCPRO 50 compression)
- MXF (MPEG-2 record and playout)
- MXF (DVCPRO 25 and DVCPRO 50 playout only)

The card supports the following audio I/O types:

- AES/EBU
- Embedded
- Analog

The audio and video connections to the card are made through the rear panel of the SGI Media Server. The panel and connections are described in the following subsections.

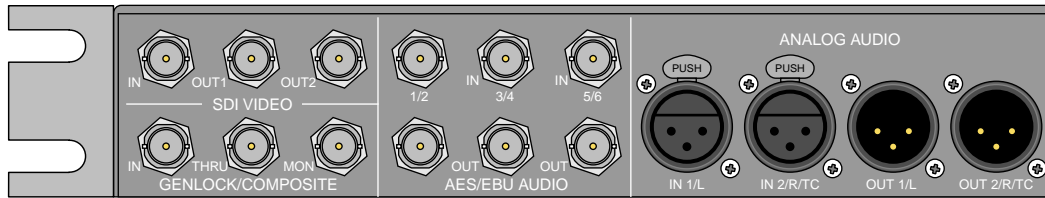
## The MSB-REARPANEL-B Rear Panel

The MSB-REARPANEL-B provides the physical audio and video connections to the SGI Media Server. A rackmounted unit supports up to two video/audio cards (channels) in the server. Each rear panel has two sets of video and audio connectors—one set for each of two audio/video cards (PCI-VIDAUD-MSB-B). Figure 2-1 shows the left-half, front view of the rear panel and shows all but one SDI video connector. The right half of the rear panel is identical to the left half with matching connectors and labeling conventions.

---

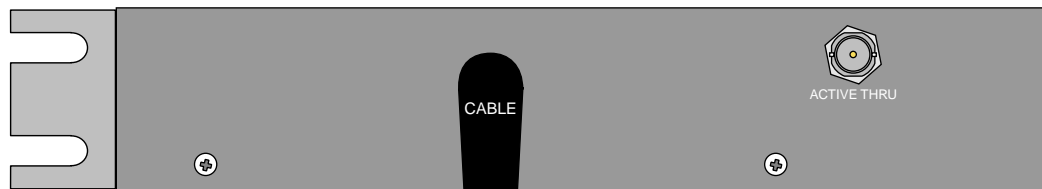
**Note:** The rackmounted rear panel requires a 19-inch connector break-out-box.

---



**Figure 2-1** MSB-REARPANEL-B Left-Half Front View

Figure 2-2 depicts the left-half, rear view and shows a single ACTIVE THRU SDI connector. As with the front of the panel, the rear halves are identical.



**Figure 2-2** MSB-REARPANEL-B Left-Half Rear View

## Video Connections

The MSB-REARPANEL provides SDI and GENLOCK/COMPOSITE video connectors.

### SDI Video Connectors

Table 2-11 provides a description of each SDI video connector.

**Table 2-11** SDI Video Connectors

---

Connector	Description
IN	This connector is the video input for the PCI-VIDAUD-MSB-B card. The input may contain a maximum eight channels of embedded digital audio conforming with the SMPTE 272M standard.
OUT1	This connector is video Output 1 for the PCI-VIDAUD-MSB card. The output may contain a maximum eight channels of embedded digital audio conforming with the SMPTE 272M standard.
OUT2	This connector, by default, is a second video output, but can also be selected to be an active loophrough of the input. The output may contain a maximum eight channels of embedded digital audio conforming with the SMPTE 272M standard
ACTIVE THRU	<p>This connector is located on the backside of the rear panel. By default, this connector is an active loophrough video output of the input but can also be selected as a second video output.</p> <p>When in active loophrough mode, the video is not processed through the PCI-VIDAUD-MSB-B card and is only reclocked within the MSB-REARPANEL-B before output.</p>

---

### Swapping Function of OUT2 and ACTIVE THRU Connectors

The modes of the OUT2 and ACTIVE THRU connectors are set by the position of an internal jumper inside the MSB-REARPANEL-B on the digital video board. By reversing the jumper's position, the function of the OUT2 and the ACTIVE-THRU connectors are swapped.

---

**Note:** All SDI video complies with SMPTE 259M specifications.

---

## Genlock/Composite Connectors

Table 2-12 describes the GENLOCK/COMPOSITE connectors.

**Table 2-12** GENLOCK/COMPOSITE Connectors

Connector	Description
IN	This connector is the Genlock video input for the PCI-VIDAUD-MSB-B card. This composite analog signal should contain either NTSC or PAL Blackburst, generated by a standard video sync signal source.
THRU	This connector is an active loopthrough output of the Genlock input. This connector may be terminated if desired. A maximum of four loopthroughs are allowed.
MON	This connector is an analog duplicate of the SDI video output of the PCI-VIDAUD-MSB-B card. The output supplies a monitor-quality signal not designed for broadcast.

## Audio Connections

The MSB-REARPANEL rear panel provides AES/EBU and analog connectors. Their capabilities are described in the following subsections.

### AES/EBU Audio In Connectors

Table 2-13 describes the AES/EBU audio input connectors.

**Table 2-13** AES/EBU Audio Input Connectors

Connector	Description
1/2	This connector is an AES/EBU digital audio input for channels 1 and 2 of the PCI-VIDAUD-MSB-B card. The input accepts two discrete channels of digital audio conforming to the SMPTE 276M standard.
3/4	This connector is an AES/EBU digital audio input for channels 3 and 4 of the PCI-VIDAUD-MSB-B card. The input accepts two discrete channels of digital audio conforming to the SMPTE 276M standard.
5/6	This connector is an AES/EBU digital audio input for channels 5 and 6 of the PCI-VIDAUD-MSB-B card. The input accepts two discrete channels of digital audio conforming to the SMPTE 276M standard.

## AES/EBU Audio Out Connectors

Table 2-14 describes the AES/EBU audio output connectors.

**Table 2-14** AES/EBU Audio Output Connectors

Connector	Description
1/2	This connector is an AES/EBU digital audio output for channels 1 and 2 of the PCI-VIDAUD-MSB-B card. The output supplies two discrete channels of digital audio conforming to the SMPTE 276M standard.
3/4	This connector is an AES/EBU digital audio output for channels 3 and 4 of the PCI-VIDAUD-MSB-B card. The output supplies two discrete channels of digital audio conforming to the SMPTE 276M standard.
5/6	This connector is an AES/EBU digital audio output for channels 5 and 6 of the PCI-VIDAUD-MSB-B card. The output supplies two discrete channels of digital audio conforming to the SMPTE 276M standard.

## ANALOG AUDIO Connectors

Table 2-15 describes the analog audio connectors.

**Table 2-15** Analog Audio Connector

Connector	Description
IN 1/L	This female XLR connector is a balanced analog audio input for channel 1 or Left of the PCI-VIDAUD-MSB-B card. The connector accepts a single audio channel.
IN 2/R/TC	This female XLR connector is a balanced analog audio input for channel 2, Right, or the SMPTE timecode of the PCI-VIDAUD-MSB-B card. It accepts a single channel of audio. A selection is available to decide if an audio channel is used when this input is used to ingest timecode. Regardless of this setting, timecode is stored as text within the index file of an MPEG-2 recording. This allows for post-processing and the availability of eight audio channels during recording.
OUT 1/L	This male XLR connector is a balanced analog audio output for channel 1 or Left of the PCI-VIDAUD-MSB-B card. The connector supplies a single audio channel.
OUT 2/4/TC	This male XLR connector is a balanced analog audio output of channel 2, Right, or the SMPTE timecode of the PCI-VIDAUD-MSB-B card. The connector supplies a single audio channel. Timecode is present on this output if previously stored as audio during recording.

## System Configuration

This chapter describes how you configure the SGI Media Server for proper operation. The chapter is organized as follows to simplify the configuration tasks:

- “Configuration Files” on page 31
- “Configuring Device Interfaces” on page 34
- “Configuring System Defaults” on page 35
- “Configuring Video Compression” on page 36
- “Configuring Audio Support” on page 39
- “Configuring the SGI Media Server for Control by Remote Devices” on page 45
- “Configuring the SGI Media Server to Control Other Devices” on page 50
- “Configuring the VTR Daemon (vtrd)” on page 52
- “Mapping Physical Ports to Logical Port Names” on page 52
- “Configuring for 625/50 Systems” on page 53
- “Tuning the SGI Media Server (Setting the Maximum DMA Size)” on page 53

---

**Note:** To configure startup options, see “Starting and Stopping VST” on page 55.

---

### Configuration Files

Table 3-1 summarizes the configuration files that the SGI Media Server uses to configure hardware and other SGI Media Server resources. These files assign controls to hardware ports that affect device and system operation.

---

**Note:** You can also use MVCP SET commands to alter control values but the changes will only persist until the next VST restart. In contrast, control values set in the configuration files persist from a VST start until changed by a SET command.

---

**Table 3-1** Configuration Files

Configuration File	Purpose
/usr/vtr/config/control-in.conf	Configures the server to be controlled by remote devices. Use this file to configure VST to work with controllers compatible with Sony and VDCP protocols and with Miranda or Horita timecode reader devices.
/usr/vtr/config/control-out.conf	Configures the server to control VTRs or VTR-like devices.
/usr/vtr/config/device-defaults	Specifies default control settings for audio, video, and disk devices.
/usr/vtr/config/ports.conf	Specifies logical video port names to substitute for the ports' physical names.
/usr/vtr/config/system-defaults	Specifies default control settings for various nondevice subsystems (for example, systemwide controls or Sony protocol control ports).
/usr/vtr/config/system-defaults/clipmirror	Specifies clip mirror configuration and a clip transmission port for the primary server. Refer to , "".
/usr/vtr/config/system-defaults/fs	Contains filesystem configuration for FTP and other persistent, storage-dependent resources. See the vst-controls(5) man page.
/usr/vtr/config/system-defaults/main	Contains information globally affecting SGI Media Server operation, such as timing standard and system event logging. See the vst-controls(5) man page.
/usr/vtr/config/system-defaults/time	Contains information regarding time management and timecode device offsets. See the vst-controls(5) man page.
/usr/vtr/config/vtrd.conf	Configures the SGI Media Server vtrd server daemon. See the vtrd(1) man page.
/usr/vtr/config/vtrdircopy.conf	Contains directory information for configuring the vtrdircopy daemon.
/usr/vtr/config/vtrftpd.conf	Contains VST control settings for managing clip transfers.



For a detailed discussion of the SGI Media Server controls and values applicable to each configuration file and device, see the `vst-control(5)` man page. The file formats and organization for the following files are described in the following sections of this chapter:

- `/usr/vtr/config/control-in.conf`
- `/usr/vtr/config/control-out.conf`
- `/usr/vtr/config/ports.conf`

Table 3-2 describes the various peer directories of `/usr/vtr/config`.

**Table 3-2** Subdirectories of `/usr/vtr`

Subdirectory	Description
<code>adm/</code>	Contains log files and crash files for <code>vvtr</code> and other executables (if crashes occur).
<code>bin/</code>	Contains commands and the VST executable <code>vvtr</code> .
<code>clips/</code>	Contains the clip cache.
<code>config/</code>	Contains configuration files.
<code>data/</code>	Contains files that are static data or image files used by various VST components.
<code>images/</code>	Contains still images created when the <code>MVCP CIMG</code> command is executed. See the <code>mvcp(5)</code> man page.
<code>index/</code>	Contains generated index files of MPEG-2 and MXF clip content stored in clips.
<code>install/</code>	Contains files used during installation.
<code>lib/perl/</code>	Contains Perl modules for application development.
<code>lib32/*.so</code>	Contains core and shared libraries.
<code>lib32/modules/*.so</code>	Contains modules for external device interfaces, controls, and formats.

## Configuring Device Interfaces

A device file contains variables called controls that map device register settings or establish settings for SGI Media Server interaction with peripheral control systems, such as the Harris Automation ADC100. The controls consist of name-value pairs. The device register settings establish specific behaviors, such as the compression format for video ingestion. Exactly one device file exists for each active media processing device in the SGI Media Server infrastructure.

### Configuring Audio and Video Devices in Your System

The directory `/usr/vtr/config/device-defaults` contains the device files corresponding to the media devices. Each device or device type has one file; the directory `device-defaults` can contain the following files:

<code>MFCODEC</code>	For behavior common to all connected multiformat codec (MFCODEC) media devices
<code>MFCODEC_n</code>	For PCI-VIDAUD-MSB-B boards and MSB-REARPANEL-B panels
<code>dq</code> or <code>dq_n</code>	For Diaquest software to control external VTRs

### Setting Device Parameters

The Video Server Technology (VST) controls are detailed in the `vst-controls(5)` man page. There are numerous controls for establishing device behavior. For example, for the first MFCODEC device, create a file under `/usr/vtr/config/device-defaults` called `MFCODEC_0`. The following shows an example of the file's contents:

```
vtr.media.video.input.compression.type mpeg2
vtr.media.mpeg.bit_rate 50000000
vtr.media.video.input.vitc.line_offset 14
vtr.media.clip.start.mode vitc
vtr.media.audio.input.channels 1
vtr.media.audio.input.channel_map.source "A"
vtr.media.audio.input.channel_map.channel_pair "1"
```

This example `device-defaults` file sets the default control values for any units created using the `MFCODEC_0` port. The example specifies the following:

- Clip recording with MPEG-2 compression at 50 megabits/second

- Detecting a vertical interval timecode (VITC) on line 14
- Processing the VITC value for each frame
- Accepting analog audio input using the MSB-REARPANEL-B connected to a PCI-VIDAUD-MSB-B peripheral

## Configuring System Defaults

You use VST controls in files in the `/usr/vtr/config/system-defaults` directory to specify default, systemwide behavior. For example, the file `main` might contain the following:

```
vtr.main.timing_standard 625
```

This line sets the default timing mode for the SGI Media Server to 625. This implies that the SGI Media Server will only process PAL-based video signals.

If a Little Red linear timecode (LTC) processor peripheral is attached, the file `/usr/vtr/config/system-defaults/time` might contain a global offset for Little-Red LTC processing:

```
vtr.time.offset -27000000
```

See the `vst-controls(5)` man page for detailed information about all system controls.

## Configuring Video Compression

The PCI-VIDAUD-MSB-B multiformat codec (MFCODEC) devices support several file formats for video compression: DIF, SGI MPEG-2, and MXF. The DIF video format supports both DVCPRO 25 and DVCPRO 50 fixed-sized compression. To enable MPEG-2 video compression, add the following line to the appropriate MFCODEC\_*n* file:

```
vtr.media.video.input.compression.type mpeg2
```

Alternatively, you can use the SET command from mvcp to specify the format directly in the unit context. Similarly, to enable DVCPRO 50 video compression, add the following line to the appropriate MFCODEC\_*n* file:

```
vtr.media.video.input.compression.type dvcpro50
```

IMX ingest requires an IMX-enabled PCI-VIDAUD-MSB-B card. The standard VST controls (see the `vst-controls(5)` man page) configure these cards for processing DVCPRO 25, DVCPRO 50, or SGI MPEG-2 video signals for playout and ingest.

To create an IMX-compatible clip, the following controls must be set within the unit context with the MVCP SET command or using entries in a device configuration file:

```
vtr.media.mpeg.bit_rate 50000000
vtr.media.video.input.constant_bit_rate true
vtr.media.input.compression.type mpeg2
vtr.media.video.input.compression.gop_structure I
vtr.media.video.input.compression.gop_size 1
```

For strict compliance with SMPTE 356M-2001, the 50 Mbps value must be used for `vtr.media.mpeg.bit_rate`. However, an IMX-compatible deck can play clips recorded at other constant bit rates.

MPEG-2 video coding can assume many compression formats, depending on the group of pictures (GOP) structure and GOP size multiplier. The PCI-VIDAUD-MSB-B MFCODEC supports four unique GOP structures: I (the default), IP, IPB, and IPBB—the standard MPEG names. For a detailed discussion of MPEG video coding, consult the website <http://www.mpeg.org>.

Each GOP structure has a GOP period, which is the interval between two successive occurrences of the GOP structure, exclusive of the I-frame. Table 3-3 defines the GOP period for each supported GOP structure.

**Table 3-3** Supported GOP Structure and Period

GOP Structure	GOP Period
I	1
IP	1
IPB	2
IPBB	3



**Caution:** Attempts to apply unsupported GOP structures may generate unpredictable results.

Each GOP structure provides variable compression encoding, though longer GOP structures typically produce higher video frame compression, thereby reducing video storage consumption during ingestion. Longer GOP structures have a drawback: they potentially introduce information loss through the encoding compression operation. The `vtr.media.mpeg.bit_rate` control, when set to a very low compression rate (less than 10000000), may contribute to information loss during encoding, though information loss depends on scene content changes and several other factors. Information loss arising from too much compression may occur during playout in the form of poor image reproduction or unwanted image artifacts.

For each supported GOP structure, the default GOP size is 1. There are many combinations of GOP size and GOP structure. These combinations are restricted by the following rule: **the GOP period must evenly divide the GOP size.**



**Caution:** Failure to obey this rule may lead to unpredictable results.

For example, Table 3-4 defines examples of valid GOP sizes, their periods, and structures.

**Table 3-4** Example GOP Size, Period, and Structure

Size	Period	Stored GOP Structure
1	1	I
15	1	IPPPPPPPPPPPPPP
6	3	IBBPBB
15	3	IBBPBBPBBPBBPBB

To enable MPEG-2 video compression with the default I-frame compression and GOP size 1, add the following line to the appropriate MFCODEC\_*n* file:

```
vtr.media.video.input.compression.type mpeg2
```

Alternatively, you can use the SET command from `mvcp` to apply it directly in the unit context. To further compress and encode a video stream, add the following lines:

```
vtr.media.video.input.compression.type mpeg2
vtr.media.video.input.compression.gop_structure IPBB
vtr.media.video.input.compression.gop_size 6
```



**Caution:** Selecting unreasonable values for

`vtr.media.video.input.compression.gop_size` may lead to unpredictable results. A maximum `gop_size` of 15 is considered an extreme upper bound for the MFCODEC.

---

The MFCODEC further supports two distinct forms of compression sampling: 4:2:0 and 4:2:2. The control `vtr.media.video.input.compression.sampling` has a default value of 422, which selects 4:2:2 compression sampling. To enable 4:2:0 compression sampling, add the following line to the appropriate MFCODEC\_*n* file:

```
vtr.media.video.input.compression.sampling 420
```

Alternatively, you can use the SET command from `mvcp` to apply it directly in the unit context.

The following controls affect only MPEG-2 compression.

- `vtr.media.video.input.compression.gop_structure`
- `vtr.media.video.input.compression.gop_size`
- `vtr.media.video.input.compression.sampling`

They are ignored for DVCPRO 25 and DVCPRO 50 video ingestion.

## Configuring Audio Support

For the MFCODEC devices, you can select the audio port map and source map for both input and output.

### Input

As shown in Table 3-5, there are four input pairs for recording. Up to four audio source selections can be made to connect to each of the input pairs. These selections are made from 8-pair or 16 audio channels, defined as follows:

- 8 channels or 4-pair of embedded input, as defined in Table 3-5
  - Embedded 1/2
  - Embedded 3/4
  - Embedded 5/6
  - Embedded 7/8
- 6 channels or 3-pair of AES/EBU input, as defined in Table 3-5
  - AES/EBU 1/2
  - AES/EBU 3/4
  - AES/EBU 5/6
- 2 channels or 1-pair analog input, as defined in Table 3-5 as Analog 1/2.

The matrix in Table 3-5 shows all possible selections.

**Table 3-5** Source Selections for Record Input Pairs

Audio Input Sources	Record Inputs of Each Channel			
	1/2	3/4	5/6	7/8
Embedded 1/2	X	X	X	X
Embedded 3/4	X	X	X	X
Embedded 5/6	X	X	X	X
Embedded 7/8	X	X	X	X
AES/EBU 1/2	X	X	X	X
AES/EBU 3/4	X	X	X	X
AES/EBU 5/6	X	X	X	X
Analog 1/2	X	X	X	X

Input notes:

- All audio selections are in pairs.
- The analog input pair can be mapped into each channel simultaneously.
- Any AES/EBU input pair can be mapped into each channel simultaneously.
- Any embedded input pair can be mapped into each channel simultaneously.
- Each channel can have only one audio source selected for it.



## Output

As shown in Table 3-6, there are four pairs (eight channels) of audio output. Each audio output type has the equivalent play output pair routed to it at all times. Analog 1/2 can be selected for simple audio monitoring.

**Table 3-6** Audio Output Mappings

Audio Output Types	Play Output Pairs			
	1/2	3/4	5/6	7/8
Embedded 1/2	X			
Embedded 3/4		X		
Embedded 5/6			X	
Embedded 7/8				X
AES/EBU 1/2	X			
AES/EBU 3/4		X		
AES/EBU 5/6			X	
Analog 1/2	X	X	X	X

### Output notes:

- Embedded audio output types are fixed to play output pairs as shown in Table 3-6.
- AES/EBU audio output types are fixed to play output pairs as shown in Table 3-6.
- The analog audio output type is selectable to any of the play output pairs.
- An “X” in Table 3-6 indicates which audio sources per channel are selected for this output.
- The play output pairs are simultaneously available for each kind of audio output types. For example, Embedded 1/2, AES/EBU 1/2, and Analog 1/2 can all utilize play output pair 1/2 at the same time.

## Channel Map/Source Control

SGI Media Server controls allow you to select what type of audio to route to each MFCODEC input pair. The choices are embedded (E), AES/EBU (D), or analog (A). The control, `vtr.media.audio.input.channel_map.source`, is set in the `/usr/vtr/config/device-defaults/MFCODEC_n` file for the unit you want to configure. This control, combined with the `vtr.media.audio.input.channel_map.channel_pair` control, selects what audio goes to each of the four MFCODEC input pairs. This can be changed on a clip-by-clip basis.

The syntax for the `vtr.media.audio.input.channel_map.source` control is as follows:

```
vtr.media.audio.input.channel_map.source value_0value_1value_2value_3
```

Each *value<sub>i</sub>* variable represents the audio type for MFCODEC input pair *i*. The possible values are E (embedded), D (AES/EBU), or A (analog). The following is an example:

```
vtr.media.audio.input.channel_map.source EEEA
```

## Channel Map/Channel Pairs Control

The `vtr.media.audio.input.channel_map.channel_pair` control selects which audio input of a particular type goes to each MFCODEC input pair. The choices are 1, 2, 3, or 4 for embedded; 1, 2, or 3 for AES/EBU; or 1 for analog. This control, combined with the `vtr.media.audio.input.channel_map.source` control, selects what audio goes to each of the four MFCODEC input pairs. This can be changed on a clip-by-clip basis.

The syntax for the `vtr.media.audio.input.channel_map.channel_pair` control is as follows:

```
vtr.media.audio.input.channel_map.channel_pair src_0src_1src_2src_3
```

Each *src<sub>i</sub>* variable represents the source choice for MFCODEC input pair *i*. The possible values are 1 (source 1), 2 (source 2), 3 (source 3), and 4 (source 4). The following is an example:

```
vtr.media.audio.input.channel_map.channel_pair 2314
```

---

**Note:** For each MFCODEC input pair, its source choice (*src\_i*) has a possible value of 1, 2, 3, or 4. However, as noted previously, these four source choices are not always available for each input type. For instance, if analog was selected for a particular MFCODEC input pair on the `vtr.media.audio.input.channel_map.source` control, the only valid value for the source is 1.

---

## Channel Map Examples

This section lists examples of audio channel mapping in the `/usr/vtr/config/device-defaults/MFCODEC_n` file.

Example 3-1 shows the default and maps embedded Inputs 1–3 to MFCODEC Inputs 0–2 and analog Input 1 (IN 1/L) to MFCODEC Input 3.

**Example 3-1** Default Audio Channel Mapping

```
vtr.media.audio.input.channel_map.source EEEA
vtr.media.audio.input.channel_map.channel_pair 1231
```

Example 3-2 maps embedded Input 1 to MFCODEC Inputs 0-1, embedded Input 3 to MFCODEC Input 2, and analog Input 1 (IN 1/L) to MFCODEC Input 3. This provides VTR-like audio control.

**Example 3-2** VTR-Like Audio Control Mapping

```
vtr.media.audio.input.channel_map.source EEEA
vtr.media.audio.input.channel_map.channel_pair 1131
```

Example 3-3 maps AES/EBU Inputs 2–3 to MFCODEC Inputs 0–1, embedded Input 1 to MFCODEC Input 2, and analog Input 1 (IN 1/L) to MFCODEC Input 3.

**Example 3-3** Mapping of AES/EBU, Embedded, and Analog Audio

```
vtr.media.audio.input.channel_map.source DDEA
vtr.media.audio.input.channel_map.channel_pair 2311
```

## Identifying Audio Ports

The `vtrhwinfo` command provides detailed information regarding the configured video and audio ports. The command lists the installed PCI-VIDAUD-MSB-B peripherals and their audio channel interfaces.

The following is sample output from the `vtrhwinfo` command:

```
Merlin Port 1 - PCI-VIDAUD-MSB Video (module=001c01 xtalk=14 slot=1)
Merlin Port 2 - PCI-VIDAUD-MSB Video (module=001c01 xtalk=14 slot=2)
Merlin Port 3 - PCI-VIDAUD-MSB Video (module=001p02 xtalk=12 slot=1)
Merlin Port 4 - PCI-VIDAUD-MSB Video (module=001p02 xtalk=12 slot=2)
Merlin Port 5 - PCI-VIDAUD-MSB Video (module=001p02 xtalk=15 slot=1)
Merlin Port 6 - PCI-VIDAUD-MSB Video (module=001p02 xtalk=15 slot=2)
```

Line 3 of the output indicates that the board (Merlin Port 3) resides in the PE brick module (denoted by the `p` in the `module` value as opposed to `c` for a compute brick), on bus 12, and in slot 1.

For example, to record two audio channels (one analog, the other embedded AES) using the `MFCODEC_14` port from the `MSB-REARPANEL-B` connected to IN 1/L and AES INPUT 1, edit the file `/usr/vtr/config/device-defaults/MFCODEC_14` and add these control settings:

```
vtr.media.audio.input.channels 2
vtr.media.audio.input.channel_map.source "AE"
vtr.media.audio.input.channel_map.channel_pair "12"
```

## Disabling Audio

If no audio inputs are connected, it is good practice to disable the controls for audio input sources and mapping. To do this, use the `MVCP SET` command within the unit context, as shown in the following session, which assumes that `U2` is created from `MFCODEC_3`:

```
UADD MFCODEC_3 * EXCL *
U2
200 OK
SET U2 MED vtr.media.audio.input.channel_map.channel_pair ""
200 OK
SET U2 MED vtr.media.audio.input.channel_map.source ""
200 OK
SET U2 MED vtr.media.audio.input.channels 0
200 OK
```

## Configuring the SGI Media Server for Control by Remote Devices

To configure the SGI Media Server so that it can be controlled by remote devices, use the `/usr/vtr/config/control-in.conf` file. The SGI Media Server supports devices that use one of the following protocols:

- Video Disk Communications Protocol (VDCP) using RS-422
- The industry-standard, Sony-compatible VTR RS-422 control protocol
- Multipoint Video Computer Protocol (MVCP) using TCP/IP

The `/usr/vtr/config/control-in.conf` file has three parts:

- Control port (serial port) configuration—one line per device
- Other control ports—one line per device
- Signal port configuration—one line per device

The control port and signal configuration lines are described in the following subsections.

### Control Port Configuration Line

You use the configuration file `/usr/vtr/config/control-in.conf` to specify basic connection parameters between the controlling devices (Sony and VDCP) and the SGI Media Server. Connection parameters include such information as which serial port controls a specific video board and the precise means of control interaction between the port and video board.

For each serial port to be connected to a controlling device (such as an automation or edit controller), insert a line in the `/usr/vtr/config/control-in.conf` file that uses the following format:

```
protocol type port speed parity signalport rate latency [clips=path/*]
```

The following line shows a sample configuration:

```
sony rs422 4 38400 1 1 29.97 0 clips=press-conference/*
```

This example specifies that a controlling device using the Sony-compatible VTR protocol is connected via an RS-422 serial line to serial port 4 on the server. The serial connection speed is 38400 bits/s, using odd parity. The device is controlling the video port identified

by the “signal 1” configuration line (also specified in `control-in.conf`) and is using a zero-frame command latency so that commands are executed as quickly as possible. The preroll should always be 0, except for VDCP controllers, for which the preroll is 3. The ingested content to be stored in the directory `/usr/vtr/clips/press-conference`.

The following sections describe each variable in this configuration line.

#### *protocol*

Use the value that corresponds to the protocol used by the device you are configuring to work with the SGI Media Server. The valid values of *protocol* include the following:

<code>sony, p2</code>	Sony-compatible VTR RS-422 control protocol
<code>louth</code>	VDCP
<code>mvcp</code>	SGI MVCP
<code>hsip</code>	Horita Serial Interface Protocol
<code>little-red</code>	Miranda Little-Red linear timecode reader protocol

#### *type*

The valid values of *type* include the following:

<code>rs232</code>	RS-232 control connection
<code>rs422</code>	RS-422 control connection
<code>tcp</code>	Socket-based TCP/IP connection

These values must correspond to the type of connection between the control video device and the SGI Media Server. The type of cable you use to connect the video device and the server determines this value.

#### *port*

The value of *port* specifies the serial port number for RS-232 and RS-422 connections. For example, to specify that the device is connected to port 2 (sometimes referred to as `ttyd2`), specify just the numeral 2. For TCP connections, *port* specifies the TCP port number.

---

**Note:** Serial and TCP ports (communications ports) are unrelated to video ports (signal ports).

---

### *speed*

The *speed* value specifies the connection speed of the serial connection between the server and the controlling device, usually 38400. On rare occasions, 9600 is used. For TCP connections, enter a hyphen (-) for this setting.

### *parity*

The optional *parity* setting can have one of the following values:

0	No parity
1	Odd parity
2	Even parity

Parity is usually odd (1). On rare occasions, even or no parity is used.

### *signalport*

The *signalport* value specifies the signal (video) port that is controlled by this remote device connection. This value is identified by a signal configuration line in the file `control-in.conf`. A prefix character of "-" implies the port is an ingest port. By default, a port is a playout port unless the port number is prefixed by "-". For more information, see "Signal Configuration Line" on page 48.

### *rate*

The *rate* value specifies the frame-rate and drop-frame mode for timecodes exchanged over the control connection. The following are the valid values:

25	625/50 video
29.97	525/59.94 video (drop-frame)
30	525/59.94 video (not drop-frame)

#### *latency*

When the SGI Media Server receives a command, the command can apply only to the next frame to enter the video port output queue. For example, if the video port output queue holds three frames, the first frame that can be affected by an incoming command is the third frame in the queue. If a stop command is received, the first two frames in the video port output queue are displayed; the third is not displayed because it is affected by the stop command.

You can modify the number of frames that display before a command begins by setting the *latency* value. Valid values are 0 or positive integers. A value of 0 starts the command's effect as soon as possible, but the command latency is always greater than 0. The minimum command latency is the sum of two values, as shown in the following:

```
minimum_command_latency = fixed_latency_value + 1
```

A small number for *latency* increases the risk that a frame may be missed (repeated) because of system load. A large number increases the latency in responding to transport commands such as play, stop, or pause.

Some commands in the Sony protocol imply a predefined frame delay; *latency* does not affect these commands.

For the MVCP TCP port, make sure that the `control-in.conf` file includes a control port configuration line:

```
mvcp tcp 5250
```

5250 is the standard MVCP control port number. The initial `control-in.conf` file installed with the VST software includes this line.

#### *clips=path/\**

This optional field specifies the directory under `/usr/vtr/clips` to contain the ingested content.

### Signal Configuration Line

The signal configuration line specifies how to map signal port numbers to video port names. A signal configuration line is required for each video port for it to be controlled



by an external device. For example, to specify that signal 1 corresponds to the video port known as MFCODEC\_4, add the following line:

```
signal 1 MFCODEC_4
```

## Example control-in.conf File

The following is a sample control-in.conf file:

```
# Control input configuration
#
# Control ports
#
# Each control port is configured with a line of this form:
#
# <prot> <comm type> <comm port> <speed> <parity> <video> <rate>
#   <preroll> <var>
#
#   prot:          Protocol (louth, odetics, sony, mvcp)
#   comm type:     rs232, rs422 (on Origin only), tcp
#   comm port:     Serial port number (e.g., 1, 2), or TCP port number
#   speed:         Bits/sec (default=38400) (use - for TCP)
#   parity:        0=none, 1=odd, 2=even (default=odd) (use - for TCP)
#   signal port:   default signal port number (use -port for louth
#                 input port)
#   frame rate:    frames/sec (29.97, 25)
#   preroll:       command preroll (# frames)
#   var:           variable arguments, currently support name=string

# This defines the normal MVCP TCP control port

mvcp tcp 5250

# Other control ports

louth  rs232 1 38400 1 2 29.97 3
louth  rs232 2 38400 1 -2 29.97 3

sony   rs422 3 38400 1 1 29.97 3

louth  rs422 5 38400 1 3 29.97 3
louth  rs422 6 38400 1 -3 29.97 3

hsip   rs232 4 9600 0 1 29.97
```

```
louth  tcp 5251 - - 4 29.97 3
louth  tcp 5252 - - -4 29.97 3

#
#
# Signal ports
#
# Each signal (video) port is mapped from a port number to port name
# with a line of this form:
#
# signal <port number> <port name>
#
# port number:      Disk port number (> 0)
# port name:       Video server port name (e.g., MFCODEC_3)

signal 1 MFCODEC_4
signal 2 MFCODEC_3
signal 3 MFCODEC_2
signal 4 MFCODEC_1
```

## Configuring the SGI Media Server to Control Other Devices

To configure the SGI Media Server server so that it can control other VTR-like devices, use the `/usr/vtr/config/control-out.conf` file.

Use this file also to specify basic connection parameters between the SGI Media Server and the VTRs to be controlled. Connection parameters include such information as which serial port controls which VTR and how. For each VTR controlled by the SGI Media Server using the Sony protocol, add a line in the `/usr/vtr/config/control-out.conf` file that uses the following format:

```
protocol    connection    serial_port    speed    parity
```

The following line shows a sample configuration:

```
sony    rs422    4    38400    1
```

The line has the same meaning when `p2` is substituted for `sony`.

The following sections describe each variable in this configuration line.

*protocol*

Use the value that corresponds to the device you are configuring to work with the server. The valid values of *protocol* are *sony* and *p2*, which are equivalent.

*connection*

The valid values of *connection* include the following:

<i>rs232</i>	RS-232 control connection
<i>rs422</i>	RS-422 control connection

These values must correspond with the type of connection between the video device and the SGI Media Server. The device and the SGI Media Server must agree on the serial connection; the cable between them must be appropriate for that connection.

*serial\_port*

The *serial\_port* value specifies the serial port as identified by a serial configuration line in the *control-in.conf* file that is controlled by this remote device connection.

Note that the RS-422 pinout at the DB-9 end of the SGI adapter cable is nonstandard. For more details, see “RS-232 and RS-422 Cabling” in Chapter 7.

*speed*

The *speed* value specifies the connection speed of the serial connection between the server and the controlling device, usually 38400. On rare occasions, 9600 is used. Sony devices must use 38400.

*parity*

The optional *parity* setting can have one of the following values:

0	No parity
1	Odd parity

## 2 Even parity

The parity must be specified as odd for the currently supported protocols.

## Configuring the VTR Daemon (`vtrd`)

The `/usr/vtr/config/vtrd.conf` file specifies information that controls the VTR daemon process (`vtrd`). The daemon process is the parent for all instances of the `vvtr` child process, the process executing for video operations. For more information about the `vtrd.conf` file format, see the `vtrd(1)` man page.

## Mapping Physical Ports to Logical Port Names

You can use the `/usr/vtr/ports.conf` file to create logical names for the physical video ports on the SGI Media Server. For each port, include a line in the configuration file with the following syntax:

```
port physicalname logicalname description
```

For example:

```
port MFCODEC_0 POTUS "WHITEHOUSE OVAL OFFICE, POTUS"  
port MFCODEC_1 MAINEB "1ST AND MAIN, Eastbound"  
port MFCODEC_5 RJM "RIJKSMUSEUM"  
port MFCODEC_6 MPEG2 "MPEG RECORDING, CAMERA 1"  
port MFCODEC_7 DVC50 "DVCPRO50 RECORDING, CAMERA 2"
```

The description field must be enclosed by quotation marks if it contains spaces. The physical port name `MFCODEC_#` references the underlying hardware. One physical port name is assigned to each installed PCI-VIDAUD-MSB codec. The physical port name is fixed at the factory. The logical port name `POTUS` may be used for unit creation and other unit-specific operations, such as changing control values using the `mcpanel` or `mcstat` command.

## Configuring for 625/50 Systems

You can configure the SGI Media Server to supports either 525/59.94 or 625/50 operation. A few operating characteristics default to 525/59.94, including the MVCP timing mode. Refer to the MVCP `FRAT` command documentation in the `mvcp(5)` man page.

To change the default timing, include the following control setting in the system defaults file `/usr/vtr/config/system-defaults/main`:

```
vtr.main.timing_standard 625
```

---

**Note:** The system timing standard controls the default video output timing only if the control `vtr.media.video.output.timing` is not changed from its default value of `system`. The `vtr.media.video.output.timing` control appears in the `/usr/vtr/config/system-defaults/MFCODEC_n` files.

---

## Tuning the SGI Media Server (Setting the Maximum DMA Size)

The maximum direct memory access (DMA) size limits the maximum amount of data that can be transferred to or from the filesystem in a single operation. The default maximum DMA size is 4 MB, which is too small for most installations. To increase the system's maximum DMA size, use the following procedure:

1. As superuser, start `sysstune` to change the DMA size:

```
# sysstune -i
```

```
Updates will be made to running system and /unix.install
sysstune->
```

2. Enter the new DMA size:

```
sysstune-> maxdmasz bytes
```

The following example shows a specification of 641 pages, which translates to approximately 10 MB (641 \* 16K):

```
sysstune-> maxdmasz 641
```

```
maxdmasz = 257 (0x101)
```

```
Do you really want to change maxdmasz to 640? (y/n)
```

3. Enter **y** for yes.

The system responds as follows:

```
In order for the change in parameter maxdmasz to become effective,  
reboot the system
```

4. Quit `systune`:

```
systune-> quit
```

5. Reboot the server:

```
# reboot
```

## Running the SGI Media Server

This chapter consists of the following topics:

- “Starting and Stopping VST” on page 55
- “Logging Events” on page 57
- “Monitoring the System” on page 61

### Starting and Stopping VST

If properly configured, the Video Server Technology (VST) software is normally started automatically when the server boots.

This section explains how to manage VST in the following sections:

- “Controlling the Autostart of VST” on page 56
- “Starting VST Manually” on page 56
- “Stopping VST” on page 56
- “Setting Startup Options” on page 57
- “Checking VST Status” on page 57

---

**Note:** For information on booting from a backup plex, see “Configuring to Boot From a Backup Plex” in Chapter 8.

---

## Controlling the Autostart of VST

By default, VST is configured to autostart when the system boots. As superuser, you can control the autostart of VST at boot time using the `chkconfig vtr` command with the `vtr` option. The following enables automatic startup:

```
# chkconfig vtr on
```

The following disables automatic startup:

```
# chkconfig vtr off
```

## Starting VST Manually

You must be `root` to start VST. Start VST with the following command:

```
# /usr/vtr/bin/vtrstart
vtrstart: Starting video server
```

---

**Note:** The message is displayed only if the system has verbose logging enabled (`chkconfig verbose on`).

---

If VST is already running, `vtrstart` displays an error:

```
vtrstart: Video server is already running (use -f to force restart)
```

As indicated, to stop the VST instance currently running and start a new VST instance, add the `-f` option:

```
# /usr/vtr/bin/vtrstart -f
vtrstart: Stopping video server
vtrstart: Starting video server
```

## Stopping VST

Stop VST with the following command:

```
# /usr/vtr/bin/vtrstop
vtrstart: Stopping video server
```



## Setting Startup Options

When the VST daemon `vtrd` starts, it reads the configuration file `/usr/vtr/config/vtrd.conf`. The daemon `vtrd` is the VST server daemon that manages the VST parent server processes.

The default installation operates normally without any change to the startup options. However, you can change the startup options for `vvtr`, the VST server process by editing the `/usr/vtr/config/vtrd.conf` file and adding them immediately after the program path, `/usr/vtr/bin/vvtr`. See the `vvtr(1)` man page for a description of the startup options.

## Checking VST Status

You can use `vtrstat` to check whether VST is running on the system. If VST is not running, `vtrstat` displays the following message:

```
# vtrstat
Video server on <host> is stopped
```

If VST is running and responding to MVCP connections, `vtrstat` displays the following:

```
Video server on <host> is running.
```

For additional VST status information displayed by `vtrstat`, see “Monitoring the System” on page 61.

## Logging Events

The VST logging feature provides a mechanism for storing a text record of events that occur during operation of the system. The default logging configuration generally records only events that are generated by various error conditions, but you can enable additional logging to help track and troubleshoot VST behavior.

This section explains the following:

- “VST Log Message Structure” on page 58
- “Configuring Logging” on page 59
- “Managing Log Rollover” on page 60

## VST Log Message Structure

Each VST log message has the following format:

```
c dd-hh:mm:ss.mmmmmmm pppp log_message
```

The fields represent the following:

*c*                    Severity code of the message  
*dd*                    Day of the month  
*hh:mm:ss.mmmmmmm*                    Time stamp that indicates when the message was written  
*pppp*                    Process ID of the process that wrote the message to the log  
*log\_message*                    Actual text message

Table 4-1 shows the VST logging levels and security codes, which are listed in decreasing order of severity.

**Table 4-1**      Logging Levels

Severity	Severity Code	Logging Level	Description
Emergency	P	-6	Panic condition.
Alert	A	-5	A condition that should be corrected immediately, such as a corrupted system file.
Critical	C	-4	A critical condition that has system-wide impact, such as a hard device error; immediate action is required.
Error	E	-3	A problem that needs correcting, but does not require immediate action.
Warning	W	-2	Possible problem, but could be a transient problem that corrects itself.
Notice	N	-1	Condition that might require attention, but is not an error condition.
Info	I	0	Informational message.
Debug	<i>n</i>	<i>n</i>	Information message that normally is of use only to engineers for debugging; can be Debug1, Debug2, or Debug3, where Debug3 produces the greatest amount of debugging information.

The following is an example of a message that has a severity code of 2 (Debug2 severity level). The message was written on day 14 of the month at the time that is shown in the message, and the ID of the process that wrote the message is 8254:

```
2 14-22:23:50.316766 8254 mvcp/ninety9 <-- 100 VTR Ready
```

The following example shows a Notice-level message:

```
N 29-09:26:11.490919 3064 U1 Unit ERROR (err=22): Clip timecode type not known
```

The following example shows an Info-level message:

```
I 29-09:26:12.758644 3065 (littlered_1) Timecode input acquired tc=09:27:35.02 (offset=* -27000000)
```

The following example shows a Warning-level message:

```
W 08-12:12:25.910495 1364 No media devices detected by "dq" module
```

## Configuring Logging

Logging is configured by setting the value of the following control in the system defaults file `/usr/vtr/config/system-defaults/main`:

```
vtr.main.log_level.file
```

For example, to enable increased logging by including the first level of debug events in the events that are logged to the server log, `/usr/vtr/adm/logs/vtrlog`, include the following control setting in the main system defaults file:

```
vtr.main.log_level.file 1
```

The value of the control is relative to the Info log event severity. As shown in Table 4-1, positive values enable more verbose logging (1 = Debug1, 2 = Debug2, ...); negative values disable all but the more severe events (-2 = Warning, -3 = Error, and so on).



**Caution:** Level 2 and above create very large files that can adversely affect system performance and should only be used for troubleshooting. Using the MVCP SET command to set the log verbosity allows you to effect the change without a VST restart.

## Managing Log Rollover

Log rollover is the practice of saving the current log file and restarting logging into an empty log file. Rollover avoids the problems of running out of disk space and of having a list of log messages too long to handle easily.

VST manages the `vtrlog` server log file. By default, the log file is saved and a new log file begun every night at 2:00 a.m. if the log file is at least 10 MB. The default number of log files retained on the system is 10.

The command `rotatelog`s is used to manage the log files and is executed by the system's cron job handler at the right time. Table 4-2 shows the list of available options.

**Table 4-2** `rotatelog`s Command Options

Option	Description	Default Value
-b	Log filename	<code>vtrlog</code>
-c	Log file compression	
-d	Log file directory	<code>/var/adm/vtr/logs</code>
-h	Command options	
-l	Daemon to notify	<code>vtrd</code>
-m	Maximum number of backups retained on the server	10
-s	Minimum log file size to trigger rollover	10 MB
-D	Debug level	Off

To change any defaults (such as the time of the rollover or the log size), the `crontab` entry of `rotatelog`s in the system must be edited (as `root`, using `crontab -e`).

For example, to retain the last 20 log files on a machine, the `rotatelog`s entry in `crontab` must look like the following:

```
1      2      *      *      *      /usr/vtr/bin/rotatelog -m 20
```

The debug option is off by default. It can be turned on using the `-D` option. There is only one level of debugging information (on).

## Monitoring the System

Once VST is installed and running, you can monitor its operation using the tools mentioned in this chapter. Two tools, `vtrstat` and `mcstat`, work directly with VST. This section describes how to monitor VST in the following subsections:

- “`vtrstat`” on page 61
- “Standard IRIX Tools” on page 62

---

**Note:** The tool `mcstat` is part of SGI Media Server control GUI, which you can use for non-production, developmental activities. The control GUI is described in Appendix A, “The SGI Media Server Control GUIs”.

---

### `vtrstat`

The tool `vtrstat` is a command-line tool that does the following:

- Indicates whether or not VST is running.
- Indicates which units are open if the `-units` option is used.
- Indicates which media ports are available if the `-ports` option is used.

Example 4-1 shows an example output of `vtrstat`.

**Example 4-1** `vtrstat` Output

```
vsta 7# /usr/vtr/bin/vtrstat -units
```

Unit	Owner	Port	Function	Status	Location	Clip
U1	mcp panel/vst12	MFCODEC_2	PLAY	RUN	16:42:39:20	dvctod
U2	mcp panel/vst12	MFCODEC_3	STOP	DONE	15:11:27:14	test1
U3	mcp panel/vst12	MFCODEC_10	STOP	DONE	15:12:29:19	test50
U4	mcp panel/vst12	MFCODEC_8	STOP	DONE	15:15:30:11	apr9/I
U5	mcp panel/vst12	MFCODEC_5	STOP	DONE	16:44:46:09	dvctod

```
vst12 8# /usr/vtr/bin/vtrstat -ports
```

#	Port	Type	Description
1	MFCODEC_0	Video	WHITEHOUSE OVAL OFFICE
2	MFCODEC_1	Video	1ST AND MAIN, E/W
3	MFCODEC_2	Video	2ND AND BROADWAY, N/S

A description of these functions can be found in Chapter 9, “Introduction to MVCP.”

## Standard IRIX Tools

You can also use the following standard SGI IRIX tools to monitor various system resources:

Tool	Description
<code>sar</code>	An activity reporter for the operating system.
<code>gr_osview</code>	A graphical system monitor. It graphically displays real-time usage of certain system resources.
<code>osview</code>	A text version of <code>gr_osview</code> .
Performance Co-Pilot (PCP)	An interface for other reporting tools. It facilitates tasks such as graphing performance data over time.

## Adding and Removing Clips

Clips are segments of audio and video. Managing clips involves adding and removing them to and from the SGI Media Server clip cache.

Managing clips is discussed in the following sections:

- “Overview of Adding Clips Procedure” on page 64
- “Media Types and Formats” on page 65
- “Adding Clips to the SGI Media Server” on page 66
- “Transferring a Clip Segment” on page 71
- “Overriding Clip Segment Transfer” on page 72
- “Removing Clips” on page 72
- “MXF Record and Playout” on page 72

## Overview of Adding Clips Procedure

This section provides a procedural overview of the tasks involved in adding clips to the SGI Media Server.

The following sections explain the steps in the following procedure in greater detail.

To add clips to the cache, use the following procedure.

1. Copy media to the SGI Media Server.

You must use a real-time filesystem. The `vtrftp` command included with Video Server Technology (VST) works in conjunction with the `vtrftpd` daemon, which is installable from the VST image. Use `get`, the familiar FTP command, with `vtrftp` to add files to the server.

For more information, see “Adding Clips to the SGI Media Server” on page 66.

2. Register the clip with the SGI Media Server.

Clips are registered automatically at VST startup. At times, however, VST may not auto detect a clip because of an unrecognized format. In that case or if you want to notify the SGI Media Server more quickly or if you have added clips to the system in a different way from those listed in step 1, use one of the following options:

- Use the Multiport Video Computer Protocol (MVCP) command `CADD`.
- Use the `vtrclip` tool.



**Caution:** The clip cache is critical to proper SGI Media Server execution. Native UNIX filesystem operations against clip cache components—specifically, clip content and index files—will corrupt (possibly permanently) the cache structure. Doing so may lead to adverse or inconsistent SGI Media Server operation. Therefore, you must use the `vtrutil` utility and the other VST commands exclusively to manipulate clip content elements outside of MVCP sessions. **DO NOT** use `rm`, `mv`, `cp`, or other native UNIX filesystem commands to manipulate clip content elements residing in the real-time filesystem.

---



## Media Types and Formats

The SGI Media Server supports two media types, *intraframe* and *interframe* media, and several media formats: DIF, SGI MPEG-2, and MXF.

### Media Types

The SGI Media Server supports intraframe media. In intraframe media, the video data for each frame is self-contained and does not depend on the data from neighboring frames. The SGI Media Server supports MPEG-2 I-frame, DVCPRO 25, and DVCPRO 50 intraframe media. The MPEG-2 DVCPRO 25 and DVCPRO 50 formats may be contained within an MXF content file.

The SGI Media Server also supports interframe media using MPEG-2 long GOP. The media type depends on the information content contained in neighboring frames of a video sequence. The SGI Media Server video codec controls parameterize the interframe compression.

### Media Formats

The SGI Media Server uses several formats for storing digital media in its clip cache filesystem(s): intraframe media in DIF format—for fixed-sized frames, including DVCPRO 25 and DVCPRO 50. Further, MPEG-2 and IMX formats may be created under appropriate hardware and VST setup. The SGI Media Server supports the following file formats:

<b>Format</b>	<b>Description</b>
SGI MPEG-2	A file format native to the SGI Media Server, which supports MPEG-2 compression, including IMX, I-frame, and long GOP.
DIF	An industry standard format used to store DVCPRO 25 and DVCPRO 50 compressed material.
MXF	An industry standard format used to store MPEG-2, DVCPRO 25, and DVCPRO 50.

## Adding Clips to the SGI Media Server

The simplest way to move clips from an editing workstation to the SGI Media Server is with FTP. VST contains a customized version of `ftpd` tailored for the VST environment and strictly dependent on a real-time filesystem. The VST version of the FTP daemon is the program `vtrftpd`. The `ftp` client is unchanged. In VST, the `inetd` process is configured to invoke `/usr/vtr/bin/vtrftpd` when a connection request from a FTP client arrives on port 20 to initiate a file transfer.

The VST FTP daemon `vtrftpd` has the following features:

- Stores to the real-time subvolume of a filesystem.
- Has configurable guaranteed rate I/O (GRIO) for stores or retrievals.
- Automatic disk- and network-I/O sizing.
- Honors in and out points while transferring DIF files.
- Updates the clip cache so that the clip is available before the end of the transfer.
- Does not quit on the end-of-file condition for growing clips.
- Can be used to set start, in, and out points of a clip.

See the `vtrftpd(1)` man page for syntax and usage options. The `vst_eoe.sw.ftpd` subsystem contains the `vtrftpd` daemon and other file transfer-related components.

## Transferring Clips to the SGI Media Server

The SGI Media Server depends on the XFS real-time filesystem, an optional component of IRIX. A real-time filesystem consists of two partitions: a data partition and the real-time partition. The real-time partition stores SGI Media Server clip content for playout and the ingested media processed during the encoding process.

Most applications, including `/usr/bsd/ftp`, have no means to distinguish among temporary filesystem characteristics. If you invoke `ftp` and use the `get` command on a SGI Media Server to get a file from another platform, the content is stored on the data partition, but not the real-time partition. However, if the `put` command is used on a remote FTP client or a passive transfer is performed, then `vtrftpd` correctly writes to the real-time partition, because `vtrftpd` is aware of the temporary filesystem.

During the `put` or passive file transfer process, `vtrftpd` periodically informs VST about clip size. This implies that an MVCP `CINF` command applied to the clip will reveal growth and change in the clip as the transfer progresses.

While the transfer is in progress, the clip is available for playout before the transfer concludes. This is useful for large clip transfers over a slow network. A transfer may also be initiated against a clip being actively recorded, thereby providing the ability to stream to another platform, such as an editing station. Unlike the standard FTP daemon `ftpd`, `vtrftpd` does not close the connection at the end of the file. The connection persists until clip growth ceases.

If the available FTP transfer bandwidth is greater than the clip compression bit rate, almost all of the clip will be available at the destination server for playout in real-time. It is also possible to daisy-chain a number of servers to concurrently stream a recorded clip over multiple hops.

The XFS real-time component provides a GRIO subsystem, which is a feature for reserving and managing filesystem resources necessary to ensure predictable response times under real-time conditions. The GRIO is necessary to avoid oversubscribing disk bandwidth. The standard FTP daemon `ftpd` consumes resources without restraint, which requires GRIO to regulate the filesystem access. Under this regulation, playout and record channels are not affected. You can specify these constraints using the controls in the `/usr/vtr/config/vtrftpd.conf` file. See the `vtrftpd(1)` man page for details.

For DIF clips, `vtrftpd` transfers only the segment within the in and out points of the clip (if they are set). This is useful when only a small portion of a large clip is needed. A clip segment can be extracted by using an FTP transfer to the local host, and then dividing a DIF clip into subclips by changing the edit points. This feature can also be disabled with the toggle command `site marks`. Consequently, the marks will be ignored and the whole clip will be transferred.

You can set the start, in, and out points of a clip using the command `site editpoints`. By default, these attributes are not transferred with the clip.

## Proxy Transfers

As mentioned previously, a `get` command with the FTP client on a server will do the wrong thing by writing the file to the data partition of your real-time filesystem. Avoid

this use of `ftp` with `vtrftpd`. Instead, use *proxy transfers* between servers, as described in the following paragraphs.

FTP allows an FTP client to control the daemon on two servers. The FTP client can be on either of the servers or on a third server. Typically, the FTP client connects both servers with a control connection, instructs one of the daemons (`vtrftpd`) to open a data connection with the other, and then initiates the transfer between the two servers. The two `vtrftpd` processes then correctly read and write the real-time files while performing GRIO reservations.

The simplest way to achieve proxy FTP transfers is to use the `vtrftp` command, a Perl FTP client that performs passive FTP transfers between servers. The `vtrftp` command copies the start, in, and out points. Enter the following command sequence to use proxy FTP with the UNIX `ftp` client:

```
server1 4# ftp server2
Connected to server2.csd.sgi.com
220 server2.csd.sgi.com FTP server ready.
Name (server2.csd:root): vtrsync
331 Password required for vtrsync.
Password:
230 User vtrsync logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> proxy open server1
Connected to server1.csd.sgi.com.
220 server1.csd.sgi.com FTP server ready.
Name (server1.csd:root): vtrsync
331 Password required for vtrsync.
Password:
230 User vtrsync logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> proxy put testftp
local: testftp remote: testftp
server1.csd.sgi.com:200 Type set to I.
server1.csd.sgi.com:227 Entering Passive Mode (130,62,124,44,8,11)
server2.csd.sgi.com:200 Type set to I.
```

```
server2.csd.sgi.com:200 PORT command successful.  
server2.csd.sgi.com:150 Opening BINARY mode data connection for  
'testftp'.  
server1.csd.sgi.com:150 Opening BINARY mode data connection for  
'testftp'  
(108969984 bytes).  
server1.csd.sgi.com:226 Transfer complete.  
server2.csd.sgi.com:226 Transfer complete.  
ftp>
```

## MPEG-2 Index File Transfer

Some SGI Media Server clip formats, including MPEG-2 clips, have both clip content and a companion index file. The index file has a clip content frame directory, consisting of metadata and frame index information essential to play the clip correctly. Both the index and clip content file must be transferred for these formats.

If the clip is being recorded and is subject to an FTP transfer to another server for playout, transfer the index and clip files simultaneously. If concurrent transfer and playout is not a requirement, transfer the index file first, and then transfer the clip content file.

Index files reside under the `/usr/vtr/index` directory and have the same name as their companion clip. Index files are small compared to the content file they describe. The `/usr/vtr/index` path is a non-real-time data partition managed by the SGI Media Server.

Use a small window size with `vtrftp`. If you use a large window size (typical for clip files), FTP will time out before the window is filled and you will get an incomplete index file.

## Clip Alignment

Certain clips might need to be aligned in the filesystem for enhanced performance. In single-disk systems, media files are aligned only with the filesystem's block size so that only one I/O operation is needed to access the data.

In multidisk, RAID striped systems, the media files must also be aligned with the stripes.

When the SGI Media Server is used to record media, it automatically places the media data in the correct, aligned locations on the disk.

### Degree of Alignment

Frame-oriented media data in an intraframe clip is aligned along the following two boundaries:

- Minor alignment boundary
- Major alignment boundary

A single element of a frame (a video field or audio chunk) never crosses a minor or major alignment boundary. The SGI Media Server will not do a read or write operation to the clip media file that crosses a major alignment boundary.

### Minor Alignment

The minor alignment matches the greater of the filesystem block size or the system memory page size. On the SGI Media Server, the minor alignment is usually 16 KB, unless a larger filesystem block size was used.

### Major Alignment

The major alignment matches the stripe size of the disk volume that holds the clip cache filesystem.

If the clip cache resides on a single disk, no major alignment is required.

If the clip cache resides on a single RAID subsystem, major alignment is required for efficient I/O access to the media data. Create a real-time filesystem on the RAID LUN and set the real-time extent size of the XFS filesystem to the desired I/O operation size.

---

**Note:** If the clip cache resides on a striped XLV volume, the major alignment matches the stripe size of the XLV volume.

---

## Notifying the SGI Media Server

The SGI Media Server detects the clips when the clip is first loaded and every time the SGI Media Server starts. The clips are then listed in `mcclips`.

If you place a clip in the clip filesystem, the first time you attempt to access the clip through the SGI Media Server, it attempts to auto-detect the clip's format and add it to the clip list. However, until you attempt to access the clip, it will not be visible in a list of clips you obtain from the Clip Manager. To see the clip immediately, use the following command:

```
# vtrclip add clip-name
```

---

**Note:** There might be a delay because adding clips always has a lower priority than `playout` or `record`.

---

You can also use the following MVCP command to notify the SGI Media Server of an added clip:

```
# CADD clip-name
```

## Transferring a Clip Segment

The `vtrftpd` utility honors in and out points on DIF clips for transfers from the real-time filesystem on the server. You can transfer the segment of a clip between in and out points instead of having to transfer the whole clip, saving transfer time and disk space in many cases.

If a clip has no in or out point set, `vtrftpd` uses the following defaults:

- If the start point (beginning of the file) is invalid or missing, it is set to 00:00:00.00.
- If the in point is invalid or missing, it is set to the start point.
- If the out point is invalid or missing, it is set to the end of the file.

## Overriding Clip Segment Transfer

If in and out points are set in a clip, but you want to transfer the entire clip, you can override this feature. To transfer an entire file that has in and out points set, enter the following at any point in the session:

```
ftp> site marks
```

This command is a toggle. To turn the clip segment transfer feature on again, enter the command again.

## Removing Clips

There are three ways to remove clips from the SGI Media Server.

- Clip Manager (mcclips)

For more information about using the Media Server Control Panel to remove a clip, see *Appendix A, "The SGI Media Server Control GUIs"*.

- Command line

You can completely remove a clip from the server using the following command:

```
# vtrclip rm clip-name
```

The value *clip-name* specifies the name of the clip file.

- MVCP command CRM.

## MXF Record and Playback

The SGI Media Server provides full record and playback support for Material Exchange Format (MXF) D-10/IMX and MPEG-2 and playback only support for DVCPRO 25 and DVCPRO 50.

Media compression formats such as MPEG-2 require a collateral frame index file to enable reliable playback and ingestion processing. This index contains a frame-by-frame description of the content clip that includes a frame directory for the content file, frame size, offset, timecode, and other pertinent information.



The following example shows truncated output of `vtrmpegutil` for the frame content of an index file:

```
param 10# vtrmpegutil -p frames /usr/vtr/index/myclip
                                index 250, pictures 250, start 01:39:00:00 (148500)
entry  offset      frame size  audio offset  audio size      Timecode
0:      0          250000 (I)    250000       30744 (0) 0 0 (0) 01:39:00:00
1:    280744      250000 (I)    530744       30744 (0) 0 0 (0) 01:39:00:01
2:    561488      250000 (I)    811488       30744 (0) 0 0 (0) 01:39:00:02
3:    842232      250000 (I)   1092232      30744 (0) 0 0 (0) 01:39:00:03
4:   1122976     250000 (I)   1372976      30744 (0) 0 0 (0) 01:39:00:04
5:   1403720     250000 (I)   1653720      30744 (0) 0 0 (0) 01:39:00:05
6:   1684464     250000 (I)   1934464      30744 (0) 0 0 (0) 01:39:00:06
7:   1965208     250000 (I)   2215208      30744 (0) 0 0 (0) 01:39:00:07
8:   2245952     250000 (I)   2495952      30744 (0) 0 0 (0) 01:39:00:08
9:   2526696     250000 (I)   2776696      30744 (0) 0 0 (0) 01:39:00:09
10:  2807440     250000 (I)   3057440      30744 (0) 0 0 (0) 01:39:00:10
```

Certain imported media formats, such as MXF, do not have native index files, and they must be generated before the SGI Media Server can play out the clip content. Playback of MXF content can be accomplished by using the `vtrmpegutil` command to generate an index file and then by invoking either `CADD clipname` from MVCP or by using the `vtrclip` command.

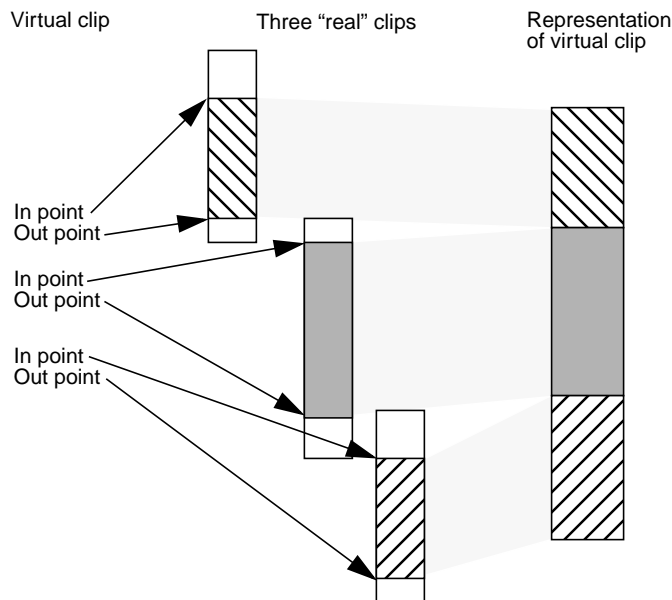
The following is an example of generating an index file for MXF content:

```
$ vtrmpegutil -i mxf clip1.mxf /usr/vtr/index/clip1.mxf
```



## Virtual Clips

A *virtual clip* (vclip) directly links clip content segments through the use of metadata. A clip content segment defines a frame-continuous portion of a larger clip, characterized by in and out points. A virtual clip consists of metadata describing the referenced content using a series of clip name references and in and out points. A virtual clip is similar to an edit decision list or a play list. When a virtual clip plays, the segments referenced by the virtual clip are seamlessly combined into a unified, continuous clip. Figure 6-1 illustrates a vclip comprised of three segments.



**Figure 6-1** Virtual Clips

A virtual clip structure is not limited to a single level of links. Further clip virtualization can be accomplished, where one virtual clip references yet another, which in turn points to physical clip content files. The format for vclip segments is unique to the SGI Media Server and is the only segment format supported by the product.

The remainder of this chapter explains the Multiport Video Computer Protocol (MVCP) commands applied for vclip operations:

- “Virtual Clip Command Overview” on page 76
- “File Operations on Virtual Clips” on page 77
- “Working with Segments” on page 78
- “Working with Frames” on page 83

## Virtual Clip Command Overview

Table 6-1 provides an alphabetical summary of the MVCP commands used to manipulate virtual clips. The remainder of the chapter describes these commands in more detail. The `mvcp(5)` man page provides detailed descriptions of the MVCP vclip commands.

**Table 6-1** MVCP Virtual Clip Commands

<b>Command</b>	<b>Description</b>
CBLD	Build a continuous clip from a virtual clip.
CCLS	Close a clip.
CFCL	Clear frames from a clip.
CFNW	Insert new frames in a clip.
CFRM	Remove frames from a clip.
CMK	Create a fetal state virtual clip.
COPN	Open a clip.
CSAV	Save a clip.
CSCL	Clear clip segments.
CSLS	List clip segments.
CSRM	Remove clip segments.
CUPS	Update clip segments.

## File Operations on Virtual Clips

This section describes the MVCP commands you use to create, open, save, close, or remove virtual clips. This section also describes how to build a non-virtual clip.

### Creating and Opening Virtual Clips

To create a virtual clip, use the MVCP command `CMK`. The syntax and arguments for `CMK` and the related command `COPN` are described briefly as follows:

```
CMK clip-name format-name
COPN clip-name
```

*clip-name* Specifies the name of the clip to create.

*format-name* Specifies the name of a segmented format. The only segmented format supported is `movie/vclip`.

`CMK` creates a fetal clip. A fetal clip is not visible in the clip cache nor can it be used for playout until clip segments are assigned using the `CUPS` command. A fetal clip has minimal state and serves as the foundation for `vclip` editing operations. The `COPN` command opens virtual or non-virtual clips for subsequent edit operations.

It is important to note that a fetal virtual clip cannot be deleted unless it contains at least one segment and has been previously closed, thus promoting it out of the fetal state into the clip cache. A fetal clip without one segment is termed *incomplete*; VST cannot process incomplete fetal clips. Attempted deletion of an incomplete fetal clip leaves an unmanageable entry, which cannot be cleared without a VST restart. Attempts to reuse a deleted, incomplete fetal clip will fail.

### Building a Non-Virtual Clip

To create a non-virtual clip, use the `CBLD` command. This command physically copies the referenced segments from a virtual clip into a continuous, non-virtual clip and assigns an arbitrary timecode for a non-virtual clip start. The syntax of the `CBLD` command is as follows:

```
CBLD clip-name new-clip [start]
```

<i>clip-name</i>	Specifies the segmented virtual clip for the extraction of segment information for the copy into the non-virtual clip destination. The segmented virtual clip must have the <code>movie/vclip</code> format.
<i>new-clip</i>	Specifies a non-virtual clip destination.
<i>start</i>	Specifies the start timecode for the non-virtual output.

## Saving, Closing, and Removing Virtual Clips

To save, close, or remove a virtual clip, use the following MVCP commands, respectively:

```
CSAV [clip-name]  
CCLS [clip-name]  
CRM [clip-name]
```

*clip-name* Specifies the name of the clip to save, close, or remove. If *clip-name* is not specified, the most recently created or opened clip is saved, closed, or removed.

---

**Note:** CSAV, CCLS, and CRM work with virtual and non-virtual clips.

---

## Working with Segments

The MVCP commands in this section work only with segment-formatted clips; currently the only format supported is `movie/vclip`.

Once you create or open a `vclip`, you add segments to or remove segments from it to revise the `vclip`.

## Adding Segments to Vclips

The CUPS (update) command, defined as follows, adds segments to `vclips`:

```
CUPS clip-name src-op dest-op trk-mask in out src-clip src-trk-mask src-in src-out
```

*clip-name* Specifies the name of the `vclip` to which you add a segment.

*src-op* Specifies the operations on the segment in the `vclip`:

- Whether the segment data (the in points and out points, for example) are copied into the vclip or simply pointed at by the vclip.
- Whether or not the segment referenced in the source clip is erased.

The following are valid values for *src-op*:

- The value `FCP` copies the segment data (the in and out points, for example) from the source clip to the target vclip, as shown in Figure 6-3.

The source clip must be segmented. If the source clip changes, the target vclip is unaffected, which is not the case with the `FLN` option.

- The value `FLN` links the segment data in the source clip to the vclip, as shown in Figure 6-3.

Unlike the `FCP` option, the source clip does not have to be segmented and if the source clip changes, its changes affect the target vclip.

- The value `FRM` copies one or more segments in the source clip to the target vclip and then remove the copied segments from the source clip.

The source clip must be segmented.

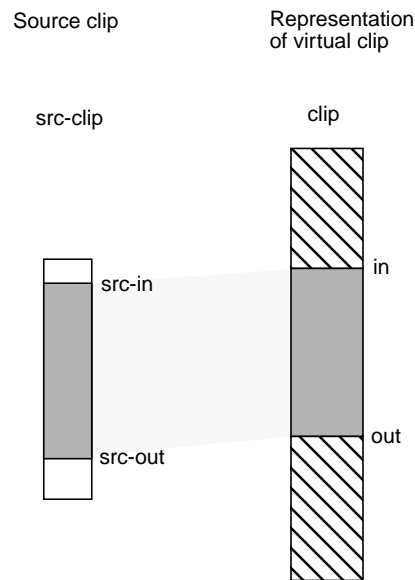
- The value `FCL` copies one or more segments in the source clip to the target vclip and then clear the copied segments from the source clip.

The source clip must be segmented. Clearing the segments, in effect, makes them black in the source clip.

<i>dest-op</i>	Specifies whether the segment added to the vclip is inserted ( <code>FINS</code> ) into the vclip or overwrites ( <code>FOVR</code> ) segments already in the vclip. Valid values are <code>FINS</code> and <code>FOVR</code> .
<i>trk-mask</i>	Must be an asterisk (*).
<i>in</i>	Specifies the in point in the vclip where the segment is to be added. If this information is absent, the new segment is appended to the end of the vclip.
<i>out</i>	Specifies the out point in the vclip where the added segment ends.
<i>src-clip</i>	Specifies the name of the source clip.
<i>src-trk-mask</i>	Must be an asterisk (*).

- src-in* Specifies the in point in the source clip. If this argument is unspecified, *src-in* is set to the in point of the source clip.
- src-out* Specifies the out point in the source clip. If this argument is unspecified, *src-out* is calculated from the following:
- The *src-in* plus *in/out* duration if the *in/out* duration is specified
  - The out point of the source clip if the *in/out* duration is not specified

Figure 6-2 illustrates some of these arguments.



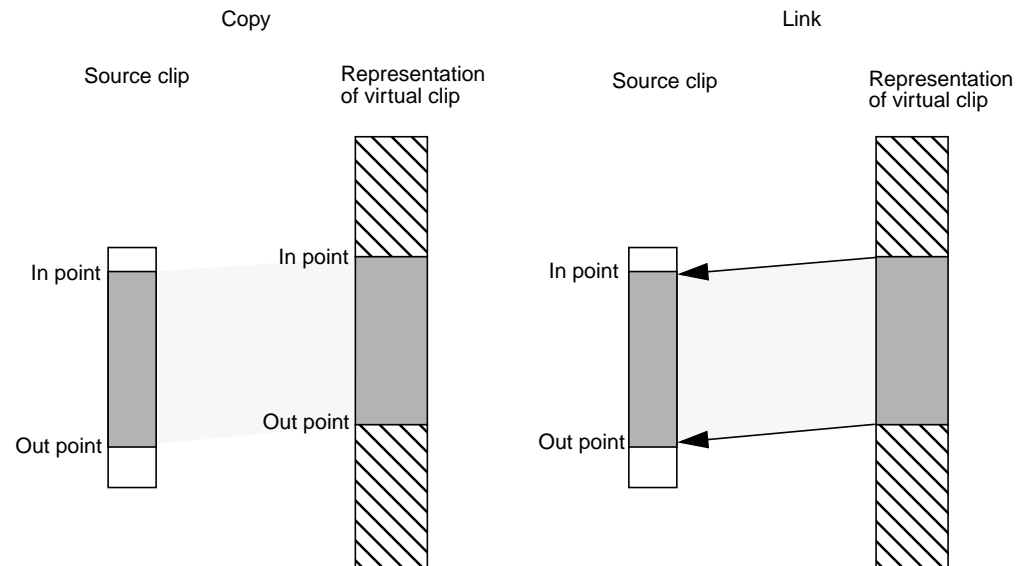
**Figure 6-2** CUPS Arguments

The interval between *src-in* and *src-out* must equal the interval between *in* and *out*; otherwise, an error is returned. If you supply only one of these intervals, CUPS automatically makes the other interval equal to it.

Using the CUPS command repeatedly enables you to populate your virtual clip with segments.



When a segment from a source clip is removed (FRM), the segments before and after the copied segment are joined. The FCL operation is not supported by the PCI-VIDAUD-MSB-B codec.



**Figure 6-3** Copying Versus Linking; Where Segment Data is Stored

## Listing Segments in a Vclip

To display pertinent information about the segments in a vclip, use the CSLS command:

```
CSLS [ clip-name [ track-mask [ in [ out ] ] ] ]
```

*clip-name* Specifies the name of the clip you are interested in. If *clip-name* is not specified, the most recently opened or created clip is used.

*track-mask* Must be an asterisk (\*).

*in* Displays information about segments beginning after this (timecode) point in the vclip. If *in* is not specified, the information displayed begins with the first segment in the vclip.

*out* Displays information about segments before this (timecode) point in the vclip. If *out* is not specified, the information displayed ends with the last segment in the vclip.

Use the *in* and *out* arguments to display information about segments in a subsection of the *vclip*.

For each segment in a *vclip*, CSL S displays a line of information in the following format:

*trk in out clip-name src-trk src-clip src-in src-out*

<i>trk</i>	Always an asterisk (*).
<i>in</i>	The in point of the segment in the <i>vclip</i> .
<i>out</i>	The out point of the segment in the <i>vclip</i> .
<i>clip-name</i>	The system-dependent name of a clip created automatically in the system used for reference counting. (This argument should be ignored.)
<i>src-trk</i>	Always an asterisk (*).
<i>src-clip</i>	The name of the clip in the clip cache where the segment comes from.
<i>src-in</i>	The in point of the segment in the source clip.
<i>src-out</i>	The out point of the segment in the source clip.

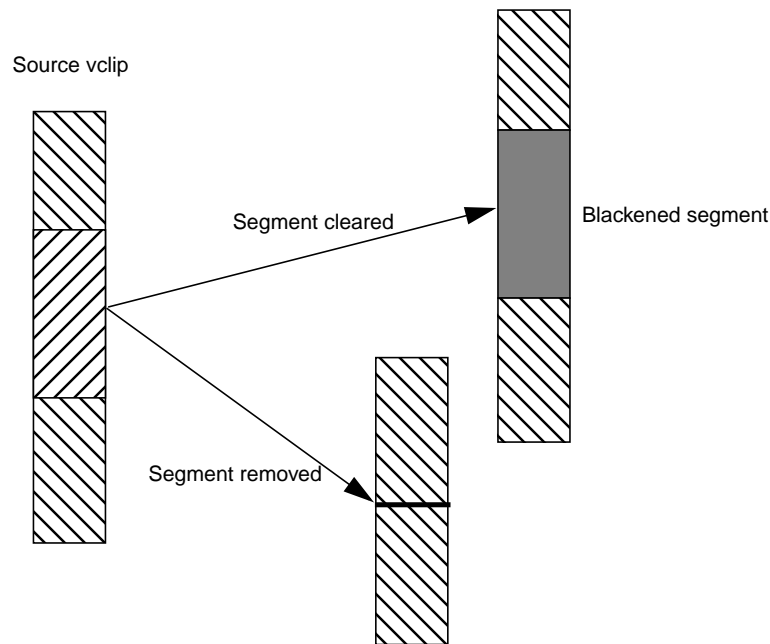
## Removing Segments in Vclips

To clear or remove segments in a virtual clip based on fixed-format clip content (DVC PRO or DVC PRO 50), use the following MVCP commands, respectively:

*CSRM clip-name track-mask timecode*

<i>clip-name</i>	Specifies the name of the clip containing the segment to be removed. If <i>clip-name</i> is not specified, the most recently created or opened clip is used.
<i>track-mask</i>	Must be an asterisk (*).
<i>timecode</i>	Specifies the timecode of the first frame of the segment you want removed.

Segment clearing using the *CSCL* command is not supported by the PCI-VIDAUD-MSB-B codec. When a segment from a specified clip is removed (*CSRM*), the segments before and after the specified segment appear contiguous, as shown in Figure 6-4.



**Figure 6-4** Clearing Versus Removing Segments

## Working with Frames

Instead of adding or removing segments, you can use MVCP commands to add or remove frames in a vclip.

### Inserting Empty Frames into a Vclip

The `CFNW` command inserts empty (black and silent) frames into a clip based on fixed-format clip content (DVCPRO or DVCPRO 50). The syntax of the `CFNW` command is as follows:

```
CFNW clip-name track-mask in out
```

*clip-name* Specifies the name of the vclip into which you want to add one or more empty frames.

<i>track-mask</i>	Must be an asterisk (*).
<i>in</i>	Specifies the in point in timecode in the vclip where the empty frames are to be inserted.
<i>out</i>	Specifies the out point in timecode in the vclip where the inserted empty frames end.

To use *CFNW*, the specified clip must have previously been opened or created using *COPN* or *CMK*, respectively, during the current MVCP session.

## Configuring and Using External Devices

Installation of the SGI Media Server includes configuring it to work with the following types of external devices:

- Devices that control the SGI Media Server. These devices include the following:
  - A VTR edit controller or other device for controlling VTRs, using a Sony-compatible (P2-compatible) RS-422 control protocol
  - An automation controller using the Video Disk Communications Protocol (VDCP)
  - Control devices using SGI Multiport Video Computer Protocol (MVCP) with TCP/IP
- Devices that the SGI Media Server controls. These include most broadcast VTRs using the Sony-compatible RS-422 control protocol.
- Devices through which media is played or recorded. These devices include PCI-VIDAUD-MSB-B cards for digital video playback and recording with DVCPRO 25, DVCPRO 50, or MPEG-2 compression format.
- Auxiliary devices, such as your timecode readers.

This chapter explains installing and configuring these devices in the following sections:

- “Configuring and Using the Sony (P2) Protocol” on page 86
- “Configuring the SGI Media Server to Control a VTR” on page 89
- “Configuring the SGI Media Server for Control by VDCP” on page 91
- “Using House Timecode Readers with the SGI Media Server” on page 96
- “RS-232 and RS-422 Cabling” on page 97

## Configuring and Using the Sony (P2) Protocol

The Sony (P2) protocol is partially supported by Video Server Technology (VST) so that the server can be controlled by standard VTR controllers. The configuration information makes the server behave similar to a video tape deck.

---

**Note:** P2 is an alternate name for the Sony protocol. For the remainder of the chapter, the term “Sony” represents both Sony and P2.

---

The following sections describe how to configure the server to work with the Sony protocol:

- “Configuring Sony Protocol Control Ports” on page 87
- “Changing the Clip Loaded in a Sony Controlled Logical Unit” on page 89



---

**Caution:** Opening a unit owned by a Sony port using MVCP commands (for example, UOPN) causes unpredictable behavior and is unsupported.

---

The SGI Media Server does not currently support the entire Sony protocol specification. It does not support the following:

- Video output parameter selection
- Audio output parameter selection
- Audio split editing
- Insert editing for anything less than all video/audio tracks

- +/-15% playback/recording/editing

---

**Note:** VAR\_FORWARD and VAR\_REVERSE are fully implemented.

---

## Configuring Sony Protocol Control Ports

To configure Sony protocol control ports, follow these steps:

1. As root, use the following command to verify that the `vst_eoe.sw.sony` subsystem is installed:

```
# /usr/vtr/bin/vtrswinfo -subsys -short
Installed software:
vst_eoe (1279037220)
  vst_eoe.man.base (1279037220)
  vst_eoe.man.ftpd (1279037220)
  vst_eoe.man.relnotes (1279037220)
  vst_eoe.man.tools (1279037220)
  vst_eoe.sw.base (1279037220)
  vst_eoe.sw.clipmirror (1279037220)
  vst_eoe.sw.diaquest (1279037220)
  vst_eoe.sw.ftpd (1279037220)
  vst_eoe.sw.little-red (1279037220)
  vst_eoe.sw.louth (1279037220)
  vst_eoe.sw.merlin (1279037220)
  vst_eoe.sw.merlin_fw (1279037220)
  vst_eoe.sw.mfcodec (1279037220)
  vst_eoe.sw.mpeg (1279037220)
  vst_eoe.sw.sony (1279037220)
  vst_eoe.sw.tools (1279037220)
```

If necessary, install the Sony subsystem.

2. Cable the controlling device to the SGI Media Server. For more information about cabling an RS-422 device to a server, see “RS-232 and RS-422 Cabling” on page 97.
3. For each Sony device controlled by the SGI Media Server, enter a line in file `/usr/vtr/config/control-in.conf`. See “Configuring the SGI Media Server for Control by Remote Devices” on page 45 for instructions.

4. Edit the control defaults file for each Sony control port as needed.

The directory `/usr/vtr/config/system_defaults` contains files specifying the default control settings for system-wide resources, including the control processors managing any Sony protocol control ports.

- Put control settings that apply to all Sony control ports in `vtr`.
- Put control settings that are limited to specific control ports in `vtr_port`, where *port* is the server serial port number to which the controlling device is attached; for example, `vtr_1`, `vtr_2`, and so on.

The following controls are supported by the Sony control module:

- `vtr.control.device_type_id`
- `vtr.control.output.idle_mode`
- `vtr.control.timecode.mode`
- `vtr.control.clip.name`
- `vtr.control.superimpose.enabled`
- `vtr.control.edit.delay`
- `vtr.control.ee.delay`
- `vtr.control.ee.mode`
- `vtr.control.ee.record_select`
- `vtr.edit.preroll`
- `vtr.edit.postroll`

See the `vst-controls(5)` man page for more information concerning the use and default values of these controls.

An application can change these controls dynamically with the MVCP `SSET` command; they can also be queried using the MVCP `SGET` command. For example:

```
SSET vtr_1 vtr.control.clip.name new-clip
```

This example creates the clip *new-clip*, if it does not already exist, and loads it onto the units controlled by the device attached to serial port 1.

---

**Note:** The setting for `vtr.control.output.idle_mode` overrides the setting for `vtr.media.output.idle_mode` of the underlying device.

---



5. Make sure that the EDIT-ON/EE-ON delay time for your edit controller matches the delay time set by VST controls.

## Changing the Clip Loaded in a Sony Controlled Logical Unit

The Sony deck control protocol was originally developed to control VTRs, thus it has no facility for clip management operations including the loading and unloading of clips. The control `vtr.control.clip.name` can be set to change the clip currently loaded.

The GUI application `mcclips` (see “The Clip Manager” in Appendix A for more information on `mcclips`) enables you to load and unload clips on ports controlled by the Sony protocol. To load a clip, for example, follow these steps:

1. Select a clip.
2. Select **File > Load On Port**.

You can also create a new clip on a port:

1. Select a clip.
2. Select **File > Create On Port**.

An application can load a new or existing clip onto the units controlled by a Sony protocol device by using the MVCP `SSET` command to set the value of `vtr.control.clip.name` (see the `vst-controls(5)` man page).

## Configuring the SGI Media Server to Control a VTR

The SGI Media Server provides frame-accurate control for a VTR or VTR-like device that supports the industry-standard Sony compatible VTR RS-422 control protocol. The VTR can be controlled interactively through a user interface or through an application for frame-accurate captures and laydowns of clips. To configure the SGI Media Server to control a VTR, follow these steps:

1. As `root`, enter the following to verify that the `vst_eoe.sw.diaquest` subsystem is installed:

```
# /usr/vtr/bin/vtrswinfo -subsys -short
```

```
Installed software:
```

```
vst_eoe (1279037220)
  vst_eoe.man.base (1279037220)
  vst_eoe.man.ftpd (1279037220)
  vst_eoe.man.relnotes (1279037220)
  vst_eoe.man.tools (1279037220)
  vst_eoe.sw.base (1279037220)
  vst_eoe.sw.clipmirror (1279037220)
  vst_eoe.sw.diaquest (1279037220)
  vst_eoe.sw.ftpd (1279037220)
  vst_eoe.sw.little-red (1279037220)
  vst_eoe.sw.louth (1279037220)
  vst_eoe.sw.merlin (1279037220)
  vst_eoe.sw.merlin_fw (1279037220)
  vst_eoe.sw.mfcodec (1279037220)
  vst_eoe.sw.mpeg (1279037220)
  vst_eoe.sw.sony (1279037220)
  vst_eoe.sw.tools (1279037220)
```

If necessary, install the Diaquest subsystem.

2. Cable the SGI Media Server to the VTR. For more information about cabling an RS-442 device to an SGI Media Server, see “RS-232 and RS-442 Cabling” on page 97.
3. For each Sony device controlled by the SGI Media Server, enter a line in file `/usr/vtr/config/control-out.conf`. See “Configuring the SGI Media Server to Control Other Devices” on page 50 for instructions.

The following example shows the `control-out.conf` configuration line for a VTR connected to serial port 3:

```
vtr rs422 3 38400 1
```

4. Edit the control defaults file as needed.

The directory `/usr/vtr/config/device-defaults` contains files specifying the default control settings for VST devices, including external VTRs controlled by the SGI Media Server.

- Put control settings that apply to all controlled VTRs control ports in `dq`.
- Put control settings that are limited to a specific controlled VTR in `dq_port`, where `port` is the VST serial port number to which the VTR is attached; for example, `dq_1`, `dq_2`, and so on.

The following are the only controls supported by the Diaquest VTR control module:

- `vtr.edit.preroll`
- `vtr.edit.postroll`

- `vtr.media.output.mode`
- `vtr.edit.coincidence.preroll`
- `vtr.media.sync_port`

To guarantee frame-accurate control, the control `vtr.media.sync_port` must be set to the name of the SGI Media Server video port that is connected to the VTR. The application `mcpnl` does this automatically, but other applications must ensure that the control is set correctly.

If the same SGI Media Server video port is always used with the VTR, you can set the value of `vtr.media.sync_port` to the name of the video port in the appropriate `dq_port` device defaults file. However, if the controlled VTR might be connected to different SGI Media Server video ports at different times, set the control using the MVCP SET command. For example, before performing a frame-accurate capture from the controlled VTR to the MFCODEC\_2 video port, set the following control:

```
SET dq-unit-name MED vtr.media.sync_port MFCODEC_2
```

For information on the controls, see the `vst-controls(5)` man page.

5. Put the VTR into remote (or slave) mode.
6. Repeat all these steps for as many VTRs as you wish to control and as are connected to the SGI Media Server.

## Configuring the SGI Media Server for Control by VDCP

The Video Disk Communications Protocol (VDCP) provides full-featured control of the SGI Media Server using an RS-422 or TCP/IP connection. The Harris Automation ADC100 controller is such a controller. Connected to the SGI Media Server, a VDCP processor supports the following:

- Back-to-back play and record (subject to restrictions imposed by the video I/O port capabilities)
- Control of multiple video (signal) ports from a single communications (control) port



**Caution:** Do not open any units (using the MVCP command `UOPEN`) belonging to a VDCP control protocol processor in order to control them.

This section consists of these subsections:

- “VDCP Commands” on page 92
- “Using the Harris Automation ADC100 Automation Controller with the SGI Media Server” on page 94

## VDCP Commands

Table 7-1 summarizes the VDCP commands that the SGI Media Server supports.

**Table 7-1** VDCP Commands Supported by the SGI Media Server

Command Type	Commands
System	Delete Protect ID UnDelete Protect ID
Immediate	Stop Play Record Still Step Continue Jog Vari. Play

**Table 7-1** VDCP Commands Supported by the SGI Media Server **(continued)**

<b>Command Type</b>	<b>Commands</b>
Reset/Select	Reset Std. Time New Copy Sort Mode Close Port Select Port Record Init Play Cue Cue with Data Delete ID Clear % to Signal Full Record Init with Data Disk Preroll
Sense Request	Open Port Next Last Port Status Request Position Request Active ID Request Device Type Request Syst. Status Request ID List ID Size Request IDs Added to Archive ID Request IDs Added List IDs Deleted List Multiport Status Request

## Using the Harris Automation ADC100 Automation Controller with the SGI Media Server

An ADC100 Harris Automation controller is connected to the SGI Media Server using one or two serial ports per the video port to be controlled. If you want to play and record at the same time, two serial port connections are required, unless you are using VDCP multiport command support.

Follow these steps to connect and configure an ADC100 Harris Automation controller:

1. Cable the automation controller to the SGI Media Server. For more information about cabling an RS-442 device to the SGI Media Server, see “RS-232 and RS-422 Cabling” on page 97.
2. Configure the automation controller. The information in Table 7-2 specifies how to configure a Harris Automation ADC100 to control the SGI Media Server. If you are using another automation controller that uses VDCP, you might need to configure that controller in a similar way.

Configure each communications port that controls one or more video ports to use the standard VDCP.

**Table 7-2** Harris Automation Device Parameters

Device Parameter	Setting
VIDEO INPUT PORT IN DISK	Video port number (corresponds to signal configuration line in file <code>control-in.conf</code> )
VIDEO OUTPUT PORT IN DISK	Video port number (corresponds to signal configuration line in file <code>control-in.conf</code> )
UPDATE EVENT DURATIONS FROM DISK	Enabled only on one port for each server
CONFIGURE INSTANT PLAY PREROLLS	Enabled, 0 seconds 4 frames
NUMBER (=) OF FRAMES TO SEND PLAY EARLY	3 frames (DVCPRO)
ENABLE BACK TO BACK PLAY	3 frames (DVCPRO)
ENABLE BACK TO BACK RECORD	3 frames (DVCPRO)
DISK HAS ARCHIVE	Enabled if an archival system is available; otherwise disabled
BACKUP PLAY FROM ARCHIVE SUPPORTED	Disabled
CACHE RECORD DISK SERIAL COMM. PORT NUMBER	0

3. For each Harris Automation communications port connected to a serial port, include a control port configuration line in the file `/usr/vtr/config/control-in.conf`. See section “Control Port Configuration Line” on page 45 for configuration instructions and examples and section “RS-232 and RS-422 Cabling” on page 97 for cabling instructions.
4. For each video port to be controlled through VDCP, include a signal port configuration line like that described in “Control Port Configuration Line” on page 45.

Example 7-1 shows the `control-in.conf` file for a configuration where VDCP is used to control four playout codecs and two input codecs. The Harris Automation communications ports that controls the playout codecs are connected to the SGI Media Server serial ports 3 through 6 and control MFCODEC\_2 through MFCODEC\_5, respectively. The communications ports controlling the ingest codecs are connected to serial ports 7 and 8 and control MFCODEC\_0 and MFCODEC\_1, respectively. Note that a “-” prefix denotes an input signal port. Both ingest ports use an input latency of 6 frames.

---

**Note:** The records containing the `louth` entries are using a legacy name required for this specific purpose.

---

**Example 7-1** Configuration File Specifications for VDCP Port Control

```
louth rs422 3 38400 1 3 29.97
louth rs422 4 38400 1 4 29.97
louth rs422 5 38400 1 5 29.97
louth rs422 6 38400 1 6 29.97
louth rs422 7 38400 1 -1 29.97 clips=path1/*
louth rs422 8 38400 1 -2 29.97 clips=path2/*

signal 1 MFCODEC_0
signal 2 MFCODEC_1
signal 3 MFCODEC_2
signal 4 MFCODEC_3
signal 5 MFCODEC_4
signal 6 MFCODEC_5
```

The configuration records contain an additional optional field, the clip pathname prefix. By default, ingested clip content is placed in directory `/usr/vtr/clips`. The clip

pathname prefix field permits you to specify an arbitrary and relative (relative to `/usr/vtr/clips`) prefix for the clip path.

As shown in Example 7-1, ports 7 and 8, corresponding to `MFCODEC_0` and `MFCODEC_1`, will have their ingested clip content placed in directories `/usr/vtr/clips/path1` and `/usr/vtr/clips/path2`, respectively. `MFCODEC_2` and `MFCODEC_3`, if used for ingest purposes, will both use the directory `/usr/vtr/clips` as the content ingest path.

## Using House Timecode Readers with the SGI Media Server

The SGI Media Server supports the Miranda Little Red and Horita PR-232 timecode. Their readers translate LTC to RS-232 timecode. Either of these devices provides a frame-accurate time reference to an external timecode generator. With such a reader attached, the SGI Media Server operations can then be synchronized with other studio equipment or with scheduled live or downlink feeds.

The timecode signal connected to the SGI Media Server is used as the reference for triggering timed-based commands for playback and recording. Frame accuracy is guaranteed within the limits defined by the specifications for the SGI Media Server.

The time-of-day signal is also used to slave the time-of-day maintained by the IRIX operating system. The time-of-day is typically maintained within one millisecond of the input time signal.

To install a timecode reader and configure the SGI Media Server for it, follow these steps:

1. Connect the timecode reader's serial port to the desired RS-232 port on the SGI Media Server.
2. Install `vst_eoe.sw.little-red`.
3. Include a control port configuration line in the file `/usr/vtr/config/control-in.conf`. See "Signal Configuration Line" on page 48 for more information. Note the following:
  - For timecode input, the signal port field ("signalport" on page 47) of the configuration line specifies the time channel number, which must be 1 for this release of the SGI Media Server.
  - Both the Miranda Little Red and Horita PR-232 timecode readers connect at a serial port speed of 9600 bits per second with no parity.



The following example shows the `control-in.conf` file for a Horita PR-232 connected to serial port 3:

```
hsip rs232 3 9600 0 1 29.97
```

This example shows a Miranda Little Red in a 625/50 configuration:

```
little-red rs232 3 9600 0 1 25
```

4. Add a line for each device to `control-in.conf` to configure the SGI Media Server to work with Horita and the Miranda Little Red timecode readers. The following is an example:

```
hsip rs232 3 9600 0 1 29.97
little-red rs232 2 9600 0 1 29.97
```

The format of the configuration line is explained in “Configuring the SGI Media Server for Control by Remote Devices” on page 45.

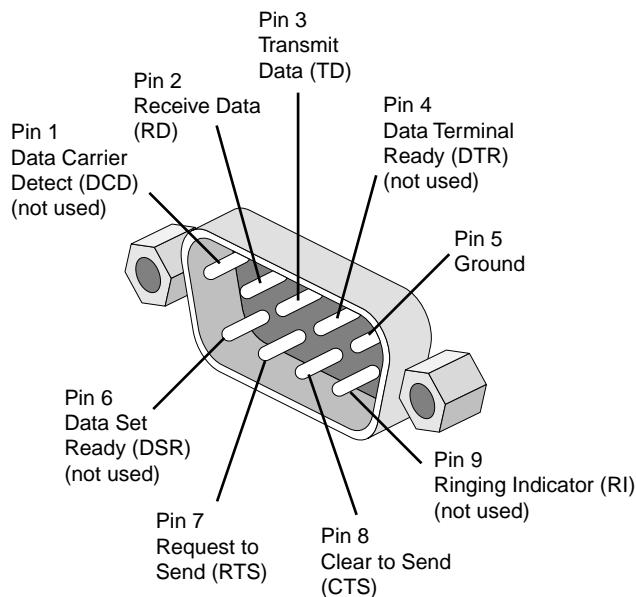
5. The `vtr.time.offset` control specifies the offset (in nanoseconds) between the actual timebase and the decoded input timecode. This offset accounts for delays in the timecode reader, in the serial communications with VST, and in VST time processing. Use the `SSET` command to change the value, or set this value in `/usr/vtr/config/system-defaults/main`. The following is an example for the Little Red:

```
SSET main vtr.time.offset -27000000
```

See <http://www.miranda.com/pdf/manual> for documentation on the Little Red device and <http://www.horita.com> for information on the PR-232.

## RS-232 and RS-422 Cabling

This section gives an overview of cabling an RS-422 connector between a Harris Automation controller and the XIO DB-9 connector on the SGI Media Server. Figure 7-1 shows the pinouts for the RS-232 and RS-422 protocols.



**Figure 7-1** Origin 300 DB-9 Pinouts for RS-232 and RS-422 Protocols

Table 7-3 describes the pin assignments for the RS-232 and RS-422 protocols.

**Table 7-3** Origin 300 DB-9 Pin Assignments

Pin	RS-232 Mode	RS-422 Mode
1	DATA CARRIER DETECT (DCD)	RESERVED
2	RECEIVE DATA (RXD)	RECEIVE DATA (RXD)
3	TRANSMIT DATA (TXD)	TRANSMIT DATA (TXD)
4	DATA TERMINAL READY (DTR)	TRANSMIT DATA+ (TXD+)
5	GROUND	SIGNAL GROUND
6	DATA SET READY (DSR)	RECEIVE DATA+ (RXD+)
7	REQUEST TO SEND (RTS)	OUTPUT HANDSHAKE (HSK <sub>o</sub> )
8	CLEAR TO SEND (CTS)	INPUT HANDSHAKE (HSK <sub>i</sub> )
9	NOT CONNECTED	RESERVED

Proper pinouts for passing RS-422 input from Mini-Din connectors to DB-9 connectors are described in the following tables. The Mini-Din connectors are those found on the Menet board. The Menet board can be identified by its six serial ports and four Ethernet ports. There are also notes for RJ12/16 and RJ45 connectors.

A proper adapter from Mini-Din to DB-9 uses the configuration shown in Table 7-4.

**Table 7-4** Configuration for a Mini-Din to DB-9 Adapter

Male 8-pin Mini-Din	Female DB-9
3 (TX-)	8 (RX-)
4 (GND)	1 (GND)
5 (RX-)	2 (TX-)
6 (TX+)	3 (RX+)
8 (RX+)	7 (TX+)

A proper adapter from DB-9 to RJ12/16 may be needed. For a Harris Automation controller, Table 7-5 shows how this adapter should look.

**Table 7-5** Adapter for Harris Automation Controller (DB-9 to RJ12/16)

DB-9 Pin (Male/Female)	Signal	RJ12/16 6-Pin
Pin 3	TXB+	Pin 1
Pin 8	TXA-	Pin 2
Pin 2	RXA-	Pin 2
Pin 7	RXB+	Pin 4
Pin 4	TX Shield	Pin 5
Pin 6	RX Shield	Pin 6

Table 7-6 shows the pin assignments for an RS-422 serial port to an RJ-45 cable connector.

**Table 7-6** Pin Assignments for an RS-422 Serial Port to an RJ-45 Cable Connector

Connector	Green Pin	Red Pin	Orange Pin	Black Pin
Origin 300 DB-9 Male	4 (TX+)	3 (TX-)	6 (RX+)	2 (RX-)
XIO MENET/MSER Board with Macintosh Mini-Din-8 to DB-9 Female Adapter	4 (TX+)	5 (TX-)	8 (RX+)	9 (RX-)
Sony (Panasonic AG-A850 or Buf RM-4000) controller DB-9 Female (Master/DCE)	7 (RXb)	2 (RXa)	3 (TXb)	8 (TXa)
Sony VTR DB-9 Female (Slave/DTE)	3 (TXb)	8 (TXa)	7 (RXb)	2 (RXa)

**Notes:**

- These pinouts are valid between an Origin system and a system not an Origin. For peer-to-peer Origin communications, configure ports as RS-232 and use a standard modem/null-modem cable. For peer-to-peer Sony communications, straight-through cables are required. Hence, they both must be slave **or** both be master.
- SGI P/N# 018-0650-001 6-inch Mini-Din 8 to DB-9 male adapter described in the XIO MENET/MSER Board Installation Instructions cannot be used for RS-422 because pin 8 (RX+ on the Mini-Din 8 end) has no corresponding pin at the DB-9 end.

## Troubleshooting

This chapter lists an assortment of common problems and their solutions.

The problems include the following:

- “625/50 Clips Do Not Play” on page 101
- “Adding a Clip Takes Excessively Long” on page 101
- “Crash: Semaphore Limit Exceeded” on page 102
- “Configuring to Boot From a Backup Plex” on page 102
- “SGI Media Server Crash Files” on page 103

### 625/50 Clips Do Not Play

The default values for the controls are NTSC-specific. If you are using a 625/50 system, set the following system control in file `/usr/vtr/config/system-defaults/main:`

```
vtr.main.timing_standard 625
```

This control sets the timing of the system, including the multiformat codec (MFCODEC) cards, to 625/50.

In the file `/usr/vtr/config/device-defaults/MFCODEC`, make sure the control `vtr.media.video.output.timing` is set to its default value, `system`.

### Adding a Clip Takes Excessively Long

Adding clips always has a lower priority than playout and recording.

## Crash: Semaphore Limit Exceeded

If `vvtr` crashes, you might have exceeded the number of configured pollable semaphores. The log would appear as follows:

```
C 06-14:29:23.348170 1386 Failed opening semaphore file descriptor
(sems open=147): No space left on device
A 06-14:29:23.348985 1386 FATAL SYSTEM ERROR:
SYSTEM CONFIGURATION ERROR
```

To fix the problem, follow these steps:

1. Edit the file `/var/sysgen/master.d/usema` and increase `USMAXDEVS`, for example, to 1050.
2. Rebuild the kernel (`autoconfig`) and reboot the system to activate the change.

## Configuring to Boot From a Backup Plex

The system normally boots from the primary root plex, that is, `root.data.0.0`. If the primary plex becomes unavailable, you can either label the disks and swap master and slave, or you can use the following procedure to set up the system so that it can boot from the secondary root plex, for example, `system.data.1.0`.

1. Ensure that `sash` (the standalone shell) is installed in the volume header of the secondary plex.
2. From the **System Maintenance Menu**, choose **Enter Command Monitor (5)**.
3. Display the PROM environment variables:

```
>> printenv
SystemPartition=dksc(0,1,8)
OSLoadPartition=dksc(0,1,0)
root=dks0d1s0
...
```

The swap PROM environment variable (which is set below) is not displayed because it is not saved in NVRAM.

4. Reset the `SystemPartition`, `OSLoadPartition`, and `root` environment variables to the values of the disk partition that contains the alternate plex and the swap environment variable to have the value of the alternate swap partition. For example:

```
>> setenv SystemPartition dksc(0,2,8)
>> setenv OSLoadPartition dksc(0,2,0)
>> setenv root dks0d2s0
>> setenv swap /dev/dsk/dks0d2s1
```

5. Exit the **Command Monitor** and restart the system:

```
>> exit
...
Option? 1
Starting up the system...
...
```

It is beyond the scope of this book to describe the intricacies of the IRIX filesystem features and tools that provide optimal performance for the SGI Media Server. For detailed information, see the latest version of *IRIX Admin: Disks and Filesystems*.

## SGI Media Server Crash Files

Inside the `/usr/vtr/adm/crash` directory is a subdirectory for each program that crashes. If a program crashes, a directory is dynamically created with that program name. Inside that directory a core file is created containing the crash information.

When the SGI Media Server detects a program crash, it renames the core file to `core.number`, where *number* is the next incremental number. This scheme prevents core files from overwriting each other.

SGI Media Server crash files are saved to aid SGI support personnel in diagnosing server software errors. Since core files may be periodically purged, generate a stack trace for the crash files to capture the pertinent information. Use the following procedure to generate a stack trace:

1. Log in to the SGI Media Server as `root`.
2. Change the directory to the appropriate core file location, as shown in the following:

```
server 1# cd /usr/vtr/adm/crash/vvtr/core.number
```

The value *number* is the core file directory of interest.

3. Run the `dbx` command on the core file as shown in the following:

```
server 2# dbx ./vvtr ./core  
.  
.  
.  
(dbx)
```

The `dbx` utility prints its header information and then prompts for more input.

4. Enter the following `dbx` command sequence:

```
(dbx)record output /usr/tmp/core.number.txt  
(dbx)where (Press the space bar or Enter key as needed.)  
(dbx)quit
```

Send the `/usr/tmp/core.number.txt` file to SGI support personnel.



## Introduction to MVCP

This chapter describes common tasks you routinely perform using MVCP commands. Each task is discussed in a tutorial fashion to introduce you to the way MVCP works. Once you master the basic tasks presented in this chapter, you can proceed to the other features offered by MVCP commands.

The tasks presented in this chapter are not logically sequential, but modular. There is a flow to the MVCP commands as presented; however, one command does not necessarily build on the one presented previous to it.

After completing a number of these tasks you will gain a feeling for the MVCP commands.

---

**Note:** For a more detailed explanation of MVCP commands, see the `mvcp(5)` man page.

---

This chapter describes the following tasks:

- “Manual Access to Video Server Technology” on page 106
- “Creating and Deleting a Unit” on page 107
- “Loading, Creating, and Unloading a Clip” on page 108
- “Finding the Name of a Clip” on page 108
- “Setting Edit Points” on page 108
- “Cueing Decks to Play or Record” on page 109
- “Sequencing Commands” on page 109
- “Playing a Prerecorded Clip” on page 109
- “Setting and Listing Configuration Values” on page 110
- “Listing Video and Deck Control Ports” on page 111
- “Assigning Audio Inputs” on page 111

- “Configuring Video Recording Compression” on page 112
- “Recording a Clip” on page 112
- “Editing Clips” on page 113
- “Displaying Your Logo” on page 114
- “Playing Clips from a Playlist” on page 115
- “Monitoring Unit State” on page 116
- “Monitoring the System” on page 117

## Manual Access to Video Server Technology

Video Server Technology (VST) applications routinely open a TCP/IP connection to a host running VST on the MVCP port, normally 5250. The application sends MVCP commands to control VST.

The port value is set in the file `/usr/vtr/config/control-in.conf`. For example, to set the port to 5250, use the following line in the file:

```
mvcp tcp 5250
```

---

**Note:** Do not change the port value from 5250 unless it is absolutely necessary.

---

You can, however, manually control VST by opening a telnet connection to a host running VST and then issuing MVCP commands to control it. For example:

```
152% telnet server 5250
Trying 130.62.156.178...
Connected to server.
Escape character is '^]'
100 VTR Ready
```

To exit the telnet session, type the following command:

**BYE**

## Creating and Deleting a Unit

A unit is a virtual VTR. It can play and record video and audio just like a VTR.

The UADD command, which you can use to create units, returns the name of the unit as shown in the following example:

```
% UADD MFCODEC_0 * SHAR
202 OK
U1
```

U1, in this case, controls the MFCODEC\_0 video board, which includes both an input video port and an output video port.

## Deleting a Unit

Units can be deleted using the following command:

```
UCLS unit-name
```

## Multiple Connections to a Unit

Once a unit has been created, it can be controlled by a number of MVCP connections by opening it:

```
UOPN unit-name
```

When multiple MVCP connections are made to one unit, the unit is not deleted until all of the connections have deleted it.



**Caution:** Opening a unit owned by a Sony or VDCP port using MVCP commands (for example, UOPN) causes unpredictable behavior.

---

## Loading, Creating, and Unloading a Clip

To load and unload a clip into a unit, use the following commands, respectively:

```
LOAD unit-name clip-name OUT CRTE  
UNLD unit-name
```

*unit-name* Specifies the name of the unit on which the clip is loaded.

*clip-name* Specifies the name of the clip to be loaded.

The OUT option means that the clip, *clip-name*, can only be played, not recorded onto. Other valid values include IN, which means the clip will be recorded onto, and BOTH, which means the clip can be played or recorded onto.

CRTE creates the clip. Without a CRTE argument, the clip does not exist in the clip cache.

## Finding the Name of a Clip

If you do not know the name of a clip, you can list all of the clips by issuing the clip list command:

```
CLS
```

## Setting Edit Points

The in and out edit points specify where the source video is to begin and end playing, and where the beginning and ending recording points are on the record deck, usually the SGI Media Server.

When a clip is played, by default the edit in point is used as the starting point for play.

To set edit points, use the CEDP command:

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

---

**Note:** The media outside of the in and out points is not actually erased; it will not be played.

---

## Cueing Decks to Play or Record

You can play and record on units without cueing them. There is, however, a good chance that the beginning of the playing or recording will not be clean; the frame count might be off by several frames or the audio might be garbled.

You can avoid these problems by cueing the units for playing or recording.

To cue a clip for playback, enter the following:

```
CUE unit-name
```

To cue a clip for recording, enter the following:

```
CUER unit-name
```

The following optional CUE and CUER arguments specify the segment of media to play, the direction of play, and how many times the segment is played:

```
CUE unit-name in-point out-point direction number-of-passes
```

Values for *direction* include FWD (forward), BWK (backward), F/B (play forward and then play backwards), and B/F (play backward and then play forward).

## Sequencing Commands

There is a variety of sequencing operations you can use, including the following:

- Sequence more than one command to execute on a unit.
- Prepend a command in the list of commands.
- Delete the command list and replace it with a command .

For information about command sequencing, see the `mvcp(5)` man page.

## Playing a Prerecorded Clip

To play a clip, use the following procedure:

1. Open a TCP/IP connection to a host machine running VST.

If successful, VST responds:

```
100 VTR Ready
```

2. Create a unit, which is a virtual videotape deck:

```
% UADD MFCODEC_0 * SHAR
```

3. Load the clip:

```
% LOAD unit-name clip-name OUT
```

*unit-name*           Name of the unit on which the clip is loaded

The OUT option means that the clip, *clip-name*, can only be played, not recorded onto.

4. Cue the clip for playback:

```
% CUE unit-name
```

5. Play the clip:

```
% PLAY unit-name speed
```

*speed*               Specifies the speed of the playback; for example, 1000 is normal play and -1000 is normal reverse play.

You can also play the clip in other modes, such as the following:

*FF*                   Fast forward.

*JOG*                 Advance or reverse one frame at a time.

*SHTL*                Variable-speed fast forward or fast reverse (shuttle).

For more information about the arguments for each of these commands, see the `mvcp(5)` man page.

## Setting and Listing Configuration Values

VST configuration variables are called *controls*. The two major categories of controls are the following:

- Device-specific** You use the SET and GET commands to specify and retrieve the control values for individual media units that control communication with the video port and audio.
- System-wide** You use the SSET and SGET commands to specify and retrieve the control values for global system values—for example, the log level.

To set or get the configuration variables for a specific unit, you have to select the subset of controls you want to set by using STOR (storage) or MED (media). For example:

```
SET unit-name MED vtr.media.video.input.compression.type dvcpro
GET unit-name MED vtr.media.video.output.*
```

These commands set the video compression type to DVCPRO 25 and return all of the video output controls for the specified unit.

## Listing Video and Deck Control Ports

To display the configuration of video and deck control ports, use the port list command:

```
PLS
```

## Assigning Audio Inputs

To identify which audio port to use by a given MFCODEC unit, use the SET command for a unit as shown in the following:

```
SET unit-name MED vtr.media.audio.input.channel_map.channel_pair "1231"
SET unit-name MED vtr.media.audio.input.channel_map.source "EEEE"
SET unit-name MED vtr.media.audio.input.channels 4
```

Alternately, you can configure a unit in `/usr/vtr/config/device-defaults/*`. For example, in `MFCODEC_1` you would enter:

```
vtr.media.audio.input.channel_map.channel_pair "1231"
vtr.media.audio.input.channel_map.source "EEEE"
vtr.media.audio.input.channels 4
```

## Configuring Video Recording Compression

To record video using the SGI Media Server, you must set the compression type using `vtr.media.video.input.compression.type`. The valid values follow:

Value	Description
<code>dvcpro</code>	For DVCPRO 25, DCT-based compression
<code>dvcpro50</code>	For DVCPRO 50, DCT-based compression
<code>mpeg2</code>	For SGI MPEG-2 long GOP, IMX, or I-frame recording

For a description of the compression types, see section “Media Formats” on page 65.

## Recording a Clip

To record a clip, use the following procedure:

1. Open a TCP/IP connection to the SGI Media Server.

If successful, the server responds:

```
100 VTR Ready
```

2. Create a unit, which is a virtual videotape deck:

```
% UADD MFCODEC_0 * SHAR
```

3. Configure VST for audio and video recording as described in sections “Assigning Audio Inputs” on page 111 and “Configuring Video Recording Compression” on page 112.

4. Load the clip:

```
% LOAD unit-name clip-name BOTH CRTE
```

*unit-name* Specifies the name of the unit on which the clip is loaded.

*clip-name* Specifies the name of the clip is to be loaded.

The BOTH option means that the clip, *clip-name*, can be played or recorded onto. The CRTE option means that the clip should be created if it does not exist.

5. Cue the clip for recording:

```
% CUER unit-name
```

6. Record onto the clip:

```
% REC unit-name
```



## Editing Clips

You can edit clips on two levels of granularity:

- Clip
- Frame

### Editing at the Clip Level

The following MVCP commands provide basic editing functionality for a clip already loaded in a unit:

CEDP            Sets in and out edit points.

*CEDP in-point out-point*

CRM            Removes a clip.

*CRM clip-name*

CCP            Copies a clip.

*CCP clip-name new-clip-name*

CMV            Renames a clip.

*CMV clip-name new-clip-name*

CLN            Creates a new clip that shares its contents with an existing clip.

*CLN clip-name new-clip-name*

When using CLN, changing one clip also changes the renamed clip because of their link.

For more information about setting in and out editing points, see “Setting Edit Points” on page 108.

## Editing at the Frame Level

The following MVCP commands manipulate the frames within a clip that is already loaded in a unit:

FRM                Removes a frame. In film, the equivalent is cutting out frames of film and splicing the two parts of the film back together.

*FRM unit-name in-frame out-frame*

FCLR               Changes (clears) a frame to black but does not remove it.

*FCLR unit-name in-frame out-frame*

FNEW               Inserts a black frame into the clip.

*FNEW unit-name in-frame out-frame*

FOVR               Moves frames from one part of a clip and overwrites frames in another part of the clip.

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

FINS               Moves frames from one part of a clip and inserts them in another part of the clip.

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

## Displaying Your Logo

You can use the MVCP SET command with the following controls to display your company logo, or any image, when the video port is not playing a clip:

```
vtr.media.output.idle_mode idle-mode  
vtr.media.video.output.image.type image-type  
vtr.media.video.output.image.name image-name
```

Place image files in directory `/usr/vtr/data/images`.

## Playing Clips from a Playlist

The commands to execute a playlist can be performed over a single MVCP control connection. Some control implementations may find it easier to use a unique MVCP control connection for each unit. The commands and the order in which they are sent is identical in either case.

1. Create a unit on the media port.

```
% UADD port
202 OK
U1
```

2. Create a second unit on the same media port.

```
% UADD port
202 OK
U2
```

3. Load the first clip into U1.

```
% LOAD U1 clip1 OUT
202 OK
clip1 movie/dif/dvcpro 108969984 108969984 00:00:00.00
00:00:30.08 * * 29.97 19990415T004204.435814Z CL
```

4. Cue U1 with a mode that will return when the command is placed in the command queue.

```
% /SEQA CUE U1
200 OK
```

5. Play U1 with a synchronization mode where a response will be returned only when the unit has started executing the command.

```
% /SEQA /SYNR PLAY U1
200 OK
```

6. Load next clip into U2.

```
% LOAD U2 clip2 OUT
202 OK
clip2 movie/dif/dvcpro 108969984 108969984 00:00:00.00
00:00:30.08 * * 29.97 19990415T004130.833005Z CL
```

7. Cue U2 with a queued command.

```
% /SEQA CUE U2
200 OK
```

8. Play U2 as with U1 above.

```
% /SEQA /SYNR PLAY U2
200 OK
```

9. Load next clip into U1.

```
% LOAD U1 clip3 OUT
202 OK
clip3 movie/dif/dvcpro 108969984 108969984 00:00:00.00
00:00:30.08 * * 29.97 19990415T004148.636950Z CL
```

10. Cue U1 with a queued command.

```
% /SEQA CUE U1
200 OK
```

11. Play U1 as above.

```
% /SEQA /SYNR PLAY U1
200 OK
```

## Monitoring Unit State

When executing a play list, you can display the state of the list execution. Using MVCP this can be accomplished by polling, but a more efficient solution is to utilize unit monitoring. Unit monitoring provides asynchronous events describing the change in state of a unit, such as execution state, loaded clips, or timecode location. Details of unit monitoring can be found in the section “Monitoring” on page 117. Typically, a separate MVCP connection will be initiated for monitoring.

In the play list algorithm in the preceding paragraph, a unit monitor could be started after the units are created to trace unit execution. This would allow a control application to trace which clip was being played and the timecode location of the unit in the clip. Note that the information provided by unit monitoring describes the unit state and locations. The actual video frame timecode leaving the server port will be some fixed number of frames behind that time due to codec delays. This delay is format-dependent and can be found in the following table. The table shows the difference between the time reported by the unit and the actual video frame at the server port at that time. For instance, with DVCPRO 25 on record the unit status reports three frames behind the

frame on the input of the server, and on play the unit status reports seven frames ahead of the frame on the server output port.

**Table 9-1** Time Reported by the Unit and the Actual Video Frame

Format	Record Unit State (frames)	Play Unit State (frames)
DVCPRO 25	-3	+7

## Monitoring the System

This section explains the following ways to monitor the system:

- “Status” on page 117
- “Monitoring” on page 117
- “Statistics” on page 118
- “Error Reporting” on page 120

### Status

You can display the status of one unit or all units using the following commands, respectively:

```
USTA unit-name
ULS
```

### Monitoring

For continuous monitoring of a unit, use the MON command:

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

*unit-name* Specifies the unit to be monitored. If you omit *unit-name*, all of the units are monitored.

*event-type* Specifies the types of events to be reported. You can receive notification when any of the following events occur on a unit in the clip cache, as appropriate:

UADD	Unit added
URM	Unit removed (using UCLS)
UCHG	Unit state changed
UCTL	Unit control changed
UERR	Unit error
CADD	Clip added
CRM	Clip removed
CEDP	Edit points changed
CCHG	Clip media changed
CMV	Clip moved
CCHP	Clip protection changed

You can add one or more of these event types to the MON command after the slash (/) that separates the unit names from the event types.

If event types are not specified, notification of UCHG, URM, and UCTL events are returned. If a unit name is not specified, notification of each UADD event is returned.

The MON command prevents an MVCP connection from processing any further commands. To stop monitoring, you must either close the MVCP connection or issue another command on the connection, at which time monitoring is terminated.

## Statistics

Two MVCP commands provide statistics:

- STLS
- STST

The following subsections describe them.

**STLS—List Statistics**

You can return statistical values for one or all components in a system using the list statistics command:

```
STLS [component-name [statistics-type]]
```

*component-name* Specifies the name of the component about which you want statistics. If omitted, statistics for all components are returned.

*statistics-type* Specifies the type of statistics you want returned. If omitted, all statistics for the specified component are returned.

If successful, the STLS command returns a line of the following form:

```
component-name statistics-type statistics-value...
```

To see the list of available statistics, type the following:

```
STLS * *
```

**STST—Statistics on Statistics**

You can return statistical values about a component or all components' statistics using the following command:

```
STST [component-name [statistics-type]]
```

For example, the following command calculates all the statistics for components that contain "MFCODEC" in their names:

```
STST *MFCODEC*
```

If the command is successful, a single line is returned in the following format:

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

*values* Specifies the number of statistical values matching the pattern.

*samples* Specifies the total number of samples collected.

*min* Specifies the minimum value.

*max* Specifies the maximum value.

*sum* Specifies the sum of the values.

<i>mean</i>	Specifies the mean of the values.
<i>stddev</i>	Specifies the standard deviation of the values.

## Error Reporting

The SGI Media Server reports three types of errors:

- MVCP command syntax errors
- Controller errors
- Unit errors

### MVCP Command Syntax Errors

All MVCP commands return error responses for syntax violations—for example, entering a letter instead of a number.

### Controller Errors

All controller and non-unit errors, such as errors in renaming or deleting a clip, return a generic notification that something failed. To receive more specific information about the most recent global error that occurred for a specific controller, use the `ERR` command:

```
ERR
```

This command returns an error code and a short description.

### Unit Errors

Unit errors are errors in the execution of unit commands, such as `PLAY`, `REC`, and `FF`. When looking at the status of a unit, you might find it in the error state, as described in “Status” on page 117. In that case, you might like to get more information about the error.

To return unit errors, use the following `UERR` command:

```
UERR unit-name
```



## Clip Mirror Configuration

A clip mirror provides an optional, hot-backup capability for the SGI Media Server. Though optional, a clip mirror platform is recommended. A clip mirror duplicates ingested media content processed by the primary SGI Media Server. This redundant structure ensures the availability of important media in the event of a catastrophic SGI Media Server failure. This chapter describes clip mirror configuration and provides guidance to allocate bandwidth for content transfers between a primary SGI Media Server and the mirror platform.

The clip mirror feature requires installation of the *clipmirror* subsystem. This subsystem contains the executable logic necessary to configure, initiate, and manage a clip mirror server. The clip mirror actively synchronizes with the primary SGI Media Server clip cache to acquire an exact copy of the media ingested by or transferred to the primary. The primary SGI Media Server normally performs media playout and record operations and may also be the principal recipient of file transfers from other video processing servers or workstations.

The SGI Media Server supports only solitary clip mirror configurations—that is, those with a one-to-one (peer-to-peer) mapping of a mirror to the primary server. Alternative clip mirror topologies are possible but are neither supported nor encouraged.

This appendix consists of the following sections:

- “Clip Mirror Service Overview” on page 122
- “Controlling Clip Mirror Operation” on page 123
- “Configuring Primary and Mirror Platform Filesystems” on page 126
- “Redesignating Servers” on page 128
- “Bandwidth Management” on page 128

## Clip Mirror Service Overview

Clip cache mirroring is performed using a TCP/IP network communications interface. The clip mirror subsystem transfers clip cache content from the primary SGI Media Server to the clip mirror platform. The transfer is unidirectional and automatic between the primary and mirror platforms and is accomplished using FTP.

The media transfer consumes both network and real-time filesystem bandwidth and, thereby, diminishes the allocated resources on the primary SGI Media Server platform. Wise allocation planning can minimize over-subscription of the real-time filesystem resource, which may adversely affect the performance of the primary SGI Media Server during playout or recording. The speed of FTP operations depends on the clip bit rate, the network infrastructure connecting the servers, and the available compute cycles to initiate the transfer.

### Automatic Clip Cache Transfers

The clip mirror service is event-driven. Certain actions performed with the primary SGI Media Server infrastructure generate clip cache activity that generates a concurrent and simultaneous action on the mirror platform. Removing clip content, creating clips, clip metadata modification, and other activities generate mirror platform actions.

Automatic clip cache transfers to the mirror platform is subject to the following specific conditions, constraints, and events:

- New content generated by or transferred to the primary SGI Media Server effects an automatic transfer.
- Upon initial clip mirror activation, if the mirror's clip content is newer (more recent modification date) than that of the primary SGI Media Server, the clip is considered synchronized; that is, no transfer takes place.
- Upon initial clip mirror activation, if the size of the clip content on the mirror is identical to that on the primary and the clip content modification date on the primary SGI Media Server is more recent, the mirror clip content is considered synchronized.
- Media content modification or metadata changes (such as protection attribute modifications) to the content on the primary SGI Media Server effects transfers.

- If content resides on the mirror platform and is absent from the primary SGI Media Server, a message is generated in `/usr/vtr/adm/logs/vtrlog` on the primary SGI Media Server.
- Clip removal or renaming on the primary SGI Media Server generates clip removal or renaming on the mirror platform. The clip may be removed manually from the mirror using an MVCP commands (CRM).

## Primary and Mirror Platform Failures

If the primary SGI Media Server becomes unavailable while being actively mirrored, the mirror platform attempts to reconnect and reestablish mirror operations. This feature persists until service to the primary SGI Media Server is restored.

In a dual-server environment, if the primary server becomes unavailable for an extended period of time, you can reconfigure the redundant server to be the primary server. When the original primary server is available again, you can reconfigure it as the new redundant server. All clips on the new primary server are then automatically copied to the newly organized and enabled mirror platform.

Clip mirroring ceases if the mirror platform fails. When the platform is restored, the service resumes; all new or changed clips will be transferred from the primary SGI Media Server. Once the clip caches on the primary SGI Media Server and mirror platform are fully synchronized, the clip mirror subsystem monitors the primary SGI Media Server for content changes to the clip cache.

## Controlling Clip Mirror Operation

The file `/usr/vtr/config/system-defaults/clipmirror` holds the control parameters specific to configuring the clip mirror subsystem. Table 10-1 lists the four

control parameters for clip mirroring. The parameters are described in the subsequent subsections.

**Table 10-1** Clip Mirror Control Parameters

Control Parameter	Meaning
<code>vtr.clipmirror.primary_server.hostname</code>	Specifies the primary SGI Media Server to mirror. Set on the clip mirror platform.
<code>vtr.clipmirror.local_server.hostname</code>	Contains the named identity of the primary SGI Media Server network interface dedicated for file transfer. Set on the primary SGI Media Server.
<code>vtr.clipmirror.max_threads</code>	Specifies the number of threads spawned on clip mirror server. Set on the clip mirror platform.
<code>vtr.clipmirror.reconnect_interval</code>	Specifies the connection retry interval if the primary SGI Media Server goes offline. Set on the clip mirror platform.

### Parameter `vtr.clipmirror.primary_server.hostname`

The `vtr.clipmirror.primary_server.hostname` control parameter must be assigned on the clip mirror platform to initiate clip mirror operations. The control value specifies the server name of the primary SGI Media Server. The following is an example file entry:

```
vtr.clipmirror.primary_server.hostname colossus.cia.gov
```

This entry establishes `colossus.cia.gov` as the primary SGI Media Server to mirror. This platform must reside within the network domain defined by the company.

An application can dynamically control clip mirroring with the MVCP `SSET` command, shown in the following example:

```
SSET clipmirror vtr.clipmirror.primary_server.hostname colossus.cia.gov
```

The `vtr.clipmirror.primary_server.hostname` control can also terminate clip mirror operations. Apply the `SSET` command with a null hostname on the mirror platform to terminate mirror activities following completion of any in-progress operations, as shown in the following example:

```
SSET clipmirror vtr.clipmirror.primary_server.hostname ""
```

**Parameter** `vtr.clipmirror.local_server.hostname`

The `vtr.clipmirror.local_server.hostname` control specifies the name of the network interface alias for transfers between the primary SGI Media Server and the mirror. A primary SGI Media Server may possess multiple network interfaces, each described by a static IP address and name alias. This control routes clip content network traffic over a specific interface from the primary SGI Media Server to the mirror platform. Clip content transfers use the default network interface configured by the primary SGI Media Server at bootstrap. Use this parameter to establish an alternate, exclusive interface for clip transfers.

Changing this control using the `SSET` command does not have an immediate effect. The clip mirror process on the SGI Media Server must be stopped and restarted using `vtrstart -f` for the new value to become active.

**Parameter** `vtr.clipmirror.max_threads`

The `vtr.clipmirror.max_threads` control parameter specifies the number of concurrent mirror operations supported by the mirror platform. The value specified by this control should be calculated from the network connection throughput and the real-time filesystem resource on the primary SGI Media Server; the default is 20 with a minimum of 1 and maximum of 100. Using a value of 1 for this control severely constrains mirror operations especially when used for high-content capture on the primary SGI Media Server.

Changing this control using the `SSET` command does not have an immediate effect. The clip mirror process on the SGI Media Server must be stopped and restarted using `vtrstart -f` for the new value to become active.

**Parameter** `vtr.clipmirror.reconnect_interval`

The `vtr.clipmirror.reconnect_interval` establishes a polling interval for connection attempts by the mirror platform to the primary SGI Media Server. The default value is 30 seconds, and 1 second is the minimum possible. Should the primary SGI Media Server cease service, the mirror platform will continue connection attempts until service resumes.

Alternatively, the mirror platform can replace the primary SGI Media Server, and the former primary SGI Media Server can be configured as a mirror platform upon service resumption.

## Configuring Primary and Mirror Platform Filesystems

The SGI Media Server filesystem requires precise tuning and configuration to provide optimal resource allocation. Therefore, it is important to match filesystem structures between the primary SGI Media Server and a mirror platform. Nonmatching filesystem structures may impede operation. The following steps ensure that the filesystems match:

1. Verify that the clip content filesystem on the mirror platform matches the real-time extent size of the primary SGI Media Server. As `root`, enter the following command:

```
# xfs_growfs -n /usr/vtr/clips
```

The output should resemble the following, which has been slightly reformatted:

```
meta-data=/usr/vtr/clips/   isize=256   agcount=8, agsize=8192 blks
data      =                  bsize=16384  blocks=65536, imaxpct=25
          =                  sunit=0      swidth=0 blks, unwritten=1
naming    =version 1        bsize=16384
log       =internal        bsize=16384  blocks=1000
realtime  =external        extsz=2097152 blocks=4445586, rtextents=34731
```

In the last line, the `extsz` value is the extent size of the filesystem.

2. On the mirror platform, use the `versions` command to verify that the following subsystems are installed:
  - `vst_eoe.sw.clipmirror`
  - `vst_eoe.sw.ftpd`
3. On the primary SGI Media Server, verify that the `vst_eoe.sw.ftpd` component is installed.
4. On the mirror platform, verify that the configuration file `/usr/vtr/config/system-defaults/clipmirror` contains clip mirror controls and values with one control specification per line. A comment line is denoted with a leading `#` character. Remove the `#` for each control desired for mirror platform configuration as described in section “Controlling Clip Mirror Operation” on page 123.

5. On the primary SGI Media Server platform, verify that the configuration file `/usr/vtr/config/system-defaults/clipmirror` contains the control `vtr.clipmirror.local_server.hostname` and has the proper value to permit transfers over a dedicated network interface.
6. On the primary SGI Media Server and mirror platform, create a new user account named `vtrsync` as shown in the following:

```
/usr/sysadm/privbin/addUserAccount -l vtrsync -u idnumber -g 0 -H
/usr/vtr/clips -s /bin/csh -P
```

The user ID (the variable `idnumber` in the example above) must be unique in the system; it can differ for different servers. To check user IDs already in the system, open the file `/etc/passwd` and view the third field. In the following example, the user ID is 994:

```
tutor::994:997:Tutorial User:/usr/tutor:/bin/csh
```

7. For the new user account `vtrsync` (on each server), set the password to `vtrsync` as shown in the following:

```
# passwd vtrsync
New Password:
Re-enter new password:
```

At each prompt shown above, enter `vtrsync`.

8. The access mode and ownership of the real-time filesystem on the primary SGI Media Server must be set to allow access by the clip mirror. On the primary SGI Media Server, enter the following:

```
# chmod -R 776 /usr/vtr/clips /usr/vtr/index
# chown root:sys /usr/vtr/clips /usr/vtr/index
```

The clip mirror service does not create subdirectories; they must be manually created on the mirror real-time filesystem as needed to match the clip directory hierarchy on the primary SGI Media Server.

## Redesignating Servers

To redesignate a mirror platform as the primary SGI Media Server, follow these steps:

1. Disable clip mirroring on the mirror platform, as shown in the following command:

```
SSET clipmirror vtr.clipmirror.primary_server.hostname " "
```

2. On a new primary SGI Media Server server, verify that the `/usr/vtr/config/system-defaults/clipmirror` file has the following control unset, as shown in the following entry:

```
vtr.clipmirror.primary_server.hostname " "
```

Unset the control if necessary.

3. On the mirror platform, set the value of the control parameter `vtr.clipmirror.primary_server.hostname` to the name of the new primary SGI Media Server.

## Bandwidth Management

Each SGI Media Server must have an XFS filesystem organized to support real-time I/O operations. This persistent storage structure must be managed by a guaranteed rate I/O (GRIO) daemon to enable predictable SGI Media Server execution. Each SGI Media Server system is parameterized by the maximum number of full-rate video streams that it supports. This constraint is based on the clip video format and the number of video codecs installed and used concurrently.

Bandwidth is consumed by the SGI Media Server for each transfer to and from an XFS real-time filesystem. Hence, bandwidth management for clip mirror operations comprises one aspect of SGI Media Server resource allocation. This bandwidth management is accomplished by limiting the total number of simultaneous transfers and the transfer rate.

The clip mirror controls and FTP controls parameterize this aspect of SGI Media Server resource allocation. The total number of simultaneous clip mirror operations is specified with `vtr.clipmirror.max_threads`.

The bandwidth consumed by FTP operations is specified by the controls contained in the `/usr/vtr/config/vtrftpd.conf` file. Table 10-2 describes these controls. The values for these controls must be specified in megabits per second.



---

**Note:** The FTP control values must match on both the primary SGI Media Server and the mirror platform.

---

**Table 10-2** FTP Control Parameters

Control Parameter	Meaning
<code>vtr.ftp.io.rate.default</code>	Specifies the default clip transfer rate (bits/s) for clip mirror operations.
<code>vtr.ftp.io.rate.enforce</code>	Specifies whether the transfer rate may exceed the value specified by the control <code>vtr.ftp.io.rate.default</code> . The value 0 allows the bandwidth to exceed that default. The value 1 enforces default transfer rate.
<code>vtr.ftp.procs.max</code>	Specifies the number of concurrent FTP transfers the primary SGI Media Server supports.

The `vtr.ftp.procs.max` control should be tuned to account for primary SGI Media Server activity, load, and workflow. The following are two possible scenarios to consider:

- Assume a primary SGI Media Server that has four video codecs with an active clip mirror. If the value of control `vtr.ftp.procs.max` is 4 and four video codecs concurrently ingest video, all four streams will be mirrored simultaneously. In this case, the values for the controls `vtr.clipmirror.max_threads` and `vtr.ftp.procs.max` should be equal.
- Assume again the same configuration of four concurrent ingest operations but also that an external source transfers content into the primary SGI Media Server. In this case, `vtr.ftp.procs.max` and `vtr.clipmirror.max_threads` should both have the following value:

$$4 + 2 * \text{number-of-external-source-transfers}$$

The `/usr/vtr/config/vtrftpd.conf` file should contain an additional record that specifies the FTP rate into and out of the clip cache path on the primary SGI Media Server. This following is an example of such an entry:

```
/usr/vtr/clips 32000000
```

This entry indicates that a maximum FTP transfer value of 32 megabits/s is allocated for clip mirroring.

The transfer rate should be slightly above the video input bit rate for the the SGI Media Server. The example above (32000000) represents approximately 1.2 times the real-time rate for a single DVCPRO 25 video stream.

The number of FTP transfers set by the `vtr.ftp.procs.max` control plus the total number of active video codecs defines the number of concurrent data streams under SGI Media Server control. This value, when multiplied by the number the value of control `vtr.ftp.io.rate.default`, determines the peak bandwidth of the XFS real-time filesystem in megabits per second. The following formula shows this symbolically:

$$\text{max-bandwidth} = (\text{vtr.ftp.procs.max} + \text{\#active\_codecs}) * \text{vtr.ftp.io.rate.default}$$

The capacity requirements for the filesystem resource can be tuned to suit this anticipated load.

## The SGI Media Server Control GUIs



---

**Caution:** Do not use the control GUIs for the control of on-air operations. The control GUIs are designed for non-production activities by developers.

---

SGI Media Server control graphical user interfaces (GUIs) consists of the following:

- “The Media Contro Panel” on page 132 (`mcpane1`), which enables clips to be played and recorded.
- “The Unit Status Monitor” on page 157 (`mcstat`), which displays the status of SGI Media Server ports.
- “The Clip Manager” on page 158 (`mcclips`), which is used to manage clips—for example, renaming, deleting, or granting permissions.

This chapter describes these tools. Additionally, the following man pages describe the GUI for the SGI Media Server:

<b>Man Page</b>	<b>Description</b>
<code>mcpa<sub>n</sub>e1(1)</code>	Describes the Media Control Panel.
<code>mcclips(1)</code>	Describes the Clip Manager.
<code>mcstat(1)</code>	Describes the status display.
<code>mccompstats(1)</code>	Describes how you monitor compression.

## Running VST Graphical Tools from a Workstation

VST is installed only on the SGI Media Server. Do not install VST or component parts on a workstation to use VST tools. To use the VST graphical tools from a workstation, do the following:

1. On your workstation, enter the following command to grant the SGI Media Server X access to your workstation:  

```
xhosts +media-server-name
```
2. Log in to the SGI Media Server.
3. Ensure that the `DISPLAY` environment variable for your server session specifies your workstation display.
4. Invoke the graphical tool.

## The Media Control Panel

This section describes how to use the Media Control Panel (`mcp-panel`) and how to control a video deck. The following topics are discussed:

- “Starting the Media Control Panel” on page 133
- “Determining Available Ports” on page 135
- “Operating the Media Control Panel” on page 138
- “Playing or Recording an Existing Clip” on page 145
- “Creating a New Clip” on page 148
- “Changing Cue Points and Edit Points” on page 149
- “Controlling a Video Deck” on page 150

See Chapter 5, “Adding and Removing Clips,” for information about how to copy digital media data from a file into the SGI Media Server clip cache.

## Starting the Media Control Panel

To establish a control connection to VST and start the Media Control Panel, enter the following:

```
% /usr/vtr/bin/mcpanel -h hostname -p videoPort|unit
```

For more complete control of `mcpanel`, use it with the following set of flags:

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

Table A-1 describes each of the options available when starting the Media Control Panel.

**Table A-1** Media Control Panel Options

Option	Description
-h	Identifies the host running VST software. If this option is not specified, the local host is assumed.
-p	Identifies the SGI Media Server video port or unit to which the control connection is made.  The value <i>videoPort</i> is the SGI Media Server video port to which the control connection is made. If you specify a port, VST creates a new logical unit that is used by this control connection.  The value <i>unit</i> is the VST 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 denoted by <i>Un</i> , for example, U4.)  If this option is not specified, the first video port on the SGI Media Server is used.
-D	Specifies the Diaquest deck control port ( <i>deckCtlPort</i> ) to be used for communication with an external video storage device, such as a digital videotape recorder (VTR). The deck is controlled by the <b>Deck Control Panel</b> .  If this option is not specified, deck control is not available.
-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.
-r	Specifies that if a control panel already exists for the video port, the existing panel should be raised on the desktop instead of creating a new one.

**Table A-1** Media Control Panel Options (**continued**)

Option	Description
-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, which means all log messages up to and including Info priority are written to <i>STDOUT</i> . (The <i>mcp-panel</i> program writes its log messages to <i>STDOUT</i> .) The values for <i>loglevel</i> are defined in Table A-2.

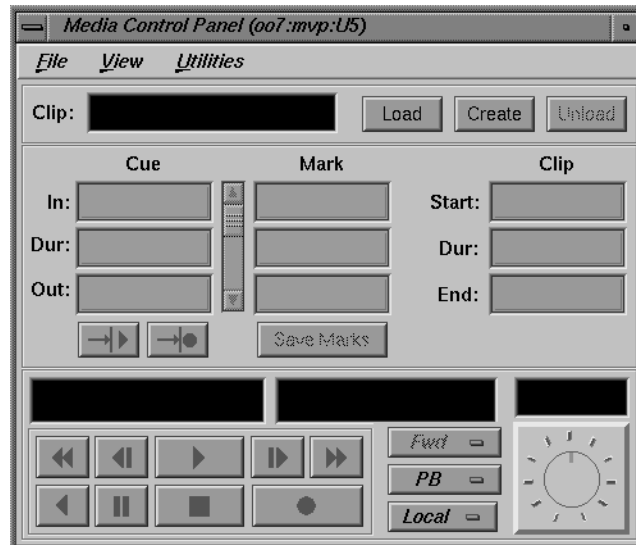
Table A-2 shows the log severity levels and codes, which are listed in decreasing order of severity.

**Table A-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 is 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 $n$	$n$	Informational message that normally is of use to engineers for debugging; may be Debug1, Debug2, or Debug3, where Debug3 produces the most debugging information.

Priority levels Info and Notice result from user actions. Priority levels Warning through Emergency generally result from system problems.

The Media Control Panel, shown in Figure A-1, is displayed in its own window. This control panel represents a unit, or logical VTR, that is used to play and record clips. The buttons in the control panel are similar in function to those of a standard VTR. For example, there are buttons to load a clip, play it, and pause.



**Figure A-1** Media Control Panel

See “Operating the Media Control Panel” on page 138 for a detailed description of the Media Control Panel.

## Determining Available Ports

To determine which video and deck control ports are available on a SGI Media Server, use the `/usr/vtr/bin/vtrstat` command as follows:

```
% /usr/vtr/bin/vtrstat -ports
# Port          Type      Description
-----
1  MFCODEC_6     Video    WHITE HOUSE OVAL OFFICE
2  MFCODEC_12    Video    RIJKSMUSEUM
```

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

### Using Telnet to Determine Available Ports

To determine which video and deck control ports are available on the SGI Media Server, establish a telnet connection to the server and then use the MVCP PLS (List Ports) command. (See section “Manual Access to Video Server Technology” on page 106 for details.)

The following example shows the PLS command output. This output identifies MFCODEC\_*n*, where *n* is 0–2, as the video ports (VID):

```
PLS
201 OK
MFCODEC_0 OUT "WHITEHOUSE OVAL OFFICE" VID MFCODEC_0
MFCODEC_1 OUT "1ST AND MAIN, E/W" VID MFCODEC_1
MFCODEC_2 OUT "2ND AND BROADWAY, N/S" VID MFCODEC_2
```

See “Listing Video and Deck Control Ports” on page 111 for more information.

### Using vtrstat to Determine Units in Use

To determine which units are in use on the SGI Media Server, use the /usr/vtr/bin/vtrstat command as follows:

```
% /usr/vtr/bin/vtrstat -units
Unit  Owner                Port          Clip          Function Location
-----
U3    mvcp/mediaserver        MFCODEC_0    *             IDLE          *
```

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

### Using Telnet to Determine Units in Use

To determine which units are in use on the SGI Media Server, establish a telnet connection to the 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 * MFCODEC_6 OUT * DONE IDLE * 0 * 0
U2 * MFCODEC_12 OUT * DONE IDLE * 0 * 0
```



## Sample Startups of the Media Control Panel

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 MFCODEC\_12 video port on the `media_server` host using a newly added unit. Messages with a severity level of Info and above are written on `STDOUT`:

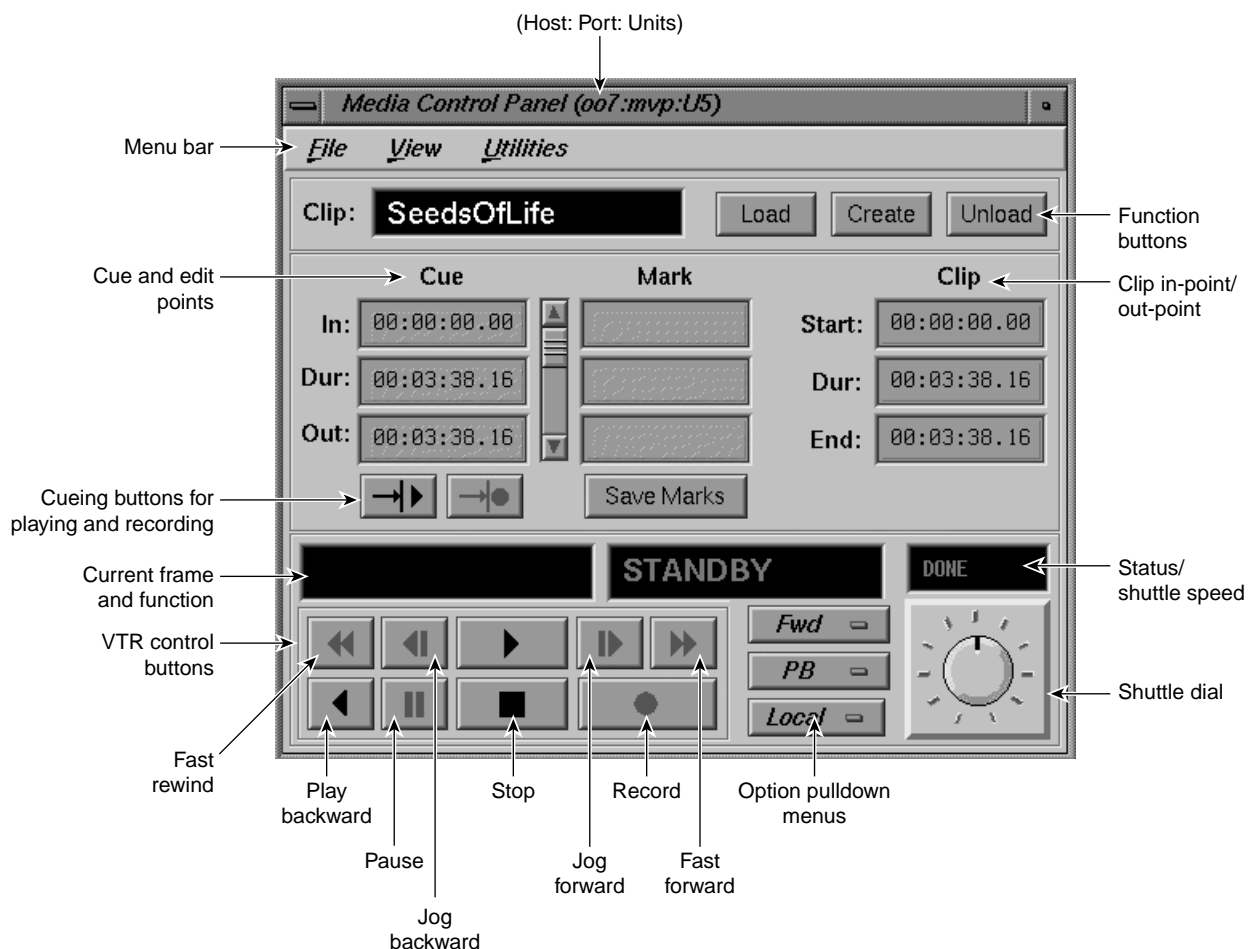
```
% /usr/vtr/bin/mcpanel -h media_server -p MFCODEC_12
```

- 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 media_server -p U9
```

## Operating the Media Control Panel

Figure A-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 A-2** 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 control panel. Most of the functions available in this pulldown menu are also available through buttons in the control panel.
- The **View** pulldown menu, which lets you access the **Deck Control** window. The Speed Control Panel, and the MPEG-2 controls. The **Deck Control** window, shown in Figure A-5 on page 151, is used to control a video deck attached to the SGI Media Server. See “Using the Deck Control Window” on page 152 for more information.
- The **Utilities** pulldown menu, which lets you access the following:
  - IRIX audio panel
  - IRIX video panel
  - Unit Status Monitor (`mcstat`), described in section “The Unit Status Monitor” on page 157
  - Clip Manager (`mcclips`), described in section “The Clip Manager” on page 158

The following describes each of the displays and buttons in the 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 (under **Cue**) 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*

The *hh* field 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 149 for information about how to change the cue points.

- The edit points (under **Mark**) 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*

The *hh* field 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 referred to as *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. The cue points are often different from the start and end points of the clip.

See "Changing Cue Points and Edit Points" on page 149 for information about how to change the edit points.

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

*hh:mm:ss:ff*

The *hh* field 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*

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 them 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 control panel's **Option** pulldown menus. (The play direction is determined by the setting of the topmost **Option** pulldown 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 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** pulldown 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:

WAIT	The function is waiting to execute or waiting for another unit to finish (for example, an automation controller that is playing a clip). The word <b>BUSY</b> blinks on and off.
RUN	The function is in progress.
DONE	The function completed without an error. The word <b>DONE</b> and the function display are grayed-out.
ERROR	An error has occurred. The word <b>ERROR</b> blinks on and off, and appears in red.

- The *VTR* control buttons control the payout and recording of clips. The following describes these buttons, which correspond to standard VTR buttons:

<<                    The fast-reverse button plays the clip in reverse at a fast speed.

- <|            The jog backward button 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.
- >             The forward play button plays the clip in the forward direction. If the clip is not cued, it is cued before it begins playout.
- |>            The Jog forward button 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.
- >>            The fast-forward button plays the clip forward at a fast speed.
- <             The reverse play button plays the clip in reverse. If the clip is not cued, it is cued before it begins playout.
- ||            The pause button temporarily stops the clip from playing or recording. You have to click the play, record, or pause button to resume.
- The stop button stops the clip and de-cues it. After the clip has been stopped, you must re-cue it before playing it again.
- The record button begins recording. If the clip is not cued, it is cued for recording before it begins.
- The top **Option** pulldown 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**            Forward play (default)
  - F Lp**            Forward loop play
  - Bwd**            Backward play
  - B Lp**            Backward loop play
  - F/Bwd**          Alternating forward and backward play
  - F/B Lp**          Alternating forward and backward loop play
  - B/Fwd**          Alternating backward and forward play
  - B/F Lp**          Alternating backward and forward loop play

**FCue** Forward play without cue (in and out) points set

**BCue** Backward play without cue (in and out) points set

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.

---

To play forward or backward without using cue points, you first use **FCue** and **BCue**, respectively, to cue the clips. When clips limits are disabled using the controls `vtr.media.clip.limit.start` and `vtr.media.clip.limit.end`, you can play forward or backward without limitation; past the beginning or end of a clip, however, the clip will play black.

- The middle **Option** pulldown 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 works 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 1 kHz 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 the SGI Media Server, even when a clip is playing. (When a clip is playing, the output displays the input signal.)

**Image**                The output always displays SMPTE 75% colors bars and plays a 1 kHz tone.

**Black**                 The output always displays a black screen.

**Hold**                  The output always displays the last image.

- The **Local/Rem Option** pulldown menu lets you put the unit in remote mode. The following options are available through this menu:

**Local**                 Puts the unit in local mode and enables the VTR control buttons (local).

**Rem**                    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 in the SGI Media Server cache appears, as shown in Figure A-3.

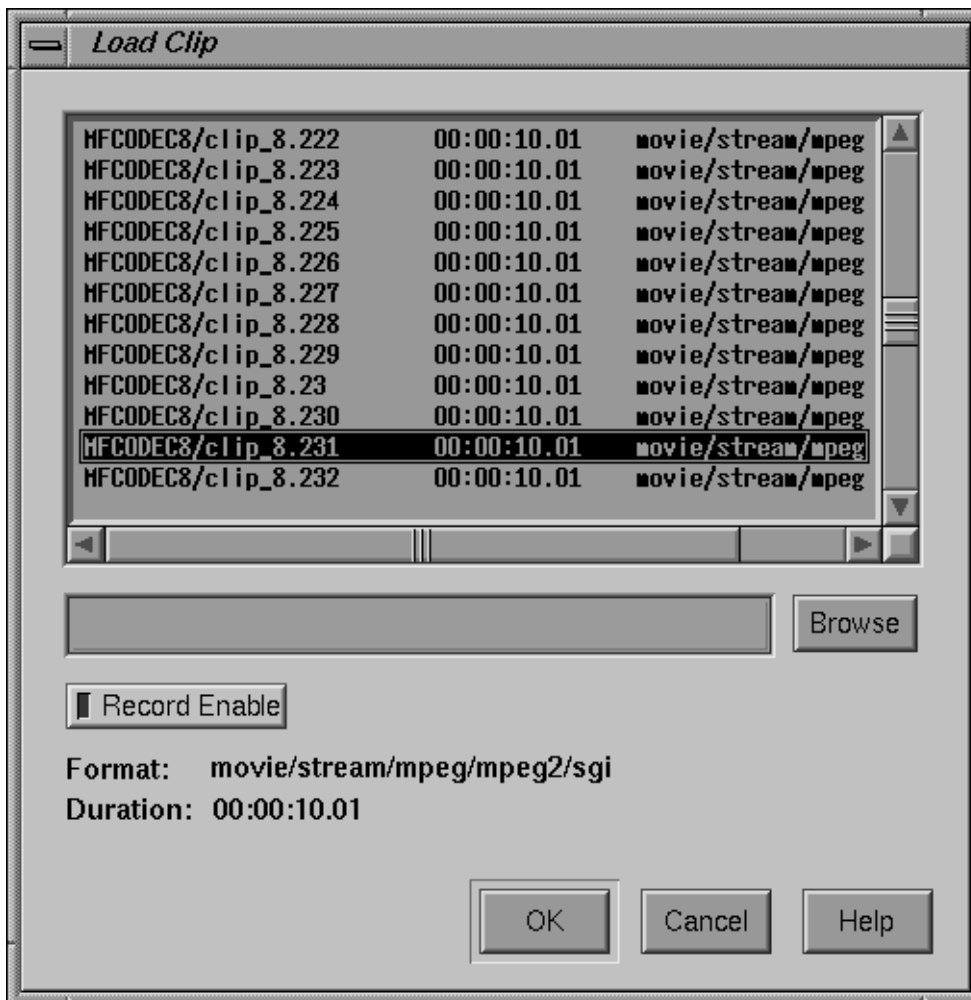


Figure A-3 Loading a Clip into the Logical VTR

The **Load Clip** window lists the name, duration, and format of each clip that is stored under `/usr/vtr/clips`, which is the VST clip cache. For example, `clip_8.231` has a duration of `00:00:10.01` and its format is MPEG-2. (If the duration contains an asterisk [\*], 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 149 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 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 (`■`).

---

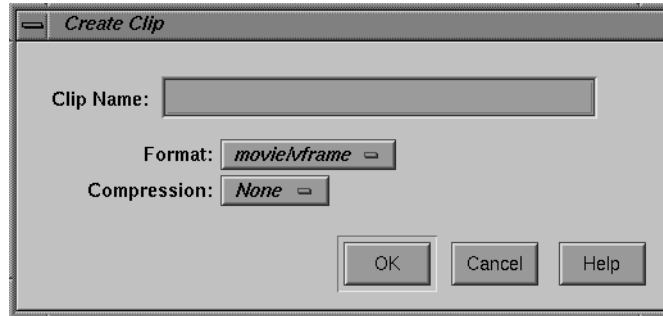
**Note:** If you want to replay or re-record the clip after it stops, the clip must be re-cued. If you click the play or record button without first clicking the cue button, the control panel automatically cues the clip before it starts playing or recording.

---

## 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 A-4, appears.



**Figure A-4** Create Clip Window

2. Enter the name of the clip in the **Clip Name** field.
3. Choose the compression type of the clip from the **Compression** option pulldown menu. By default, `mpeg2` is chosen for the compression type.
4. Choose the format of the movie from the **Format** option pulldown menu.

You can create clips through the Media Control Panel using one of these options:

- `dvcpro`
- `dvcpro50`
- `mpeg2`

5. Click the **OK** button. The window closes and the clip appears in the control panel. Both the cue for recording button ( $\rightarrow | \bullet$ ) and the cue for playout button ( $\rightarrow | >$ ) are enabled.
6. Click the cue for recording button ( $\rightarrow | \bullet$ ).
7. To start recording, click the record button ( $\bullet$ ).
8. To stop recording, click the stop button ( $\blacksquare$ ).

For more information about controlling the video deck, see “Controlling a Video Deck” on page 150.

## 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 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 control panel verifies the new value. If the new value is valid, the control panel adjusts the other values to correspond to the new one, if necessary. If the new value is invalid, the 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 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 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 the clip is cued. They do not affect a clip that is already cued or is playing.

---

Any changes 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 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 SGI Media Server. (For information on attaching a video deck to the SGI Media Server, see Chapter 7, “Configuring and Using External Devices.”)

---

**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 Media Control Panel” on page 133 for details.)

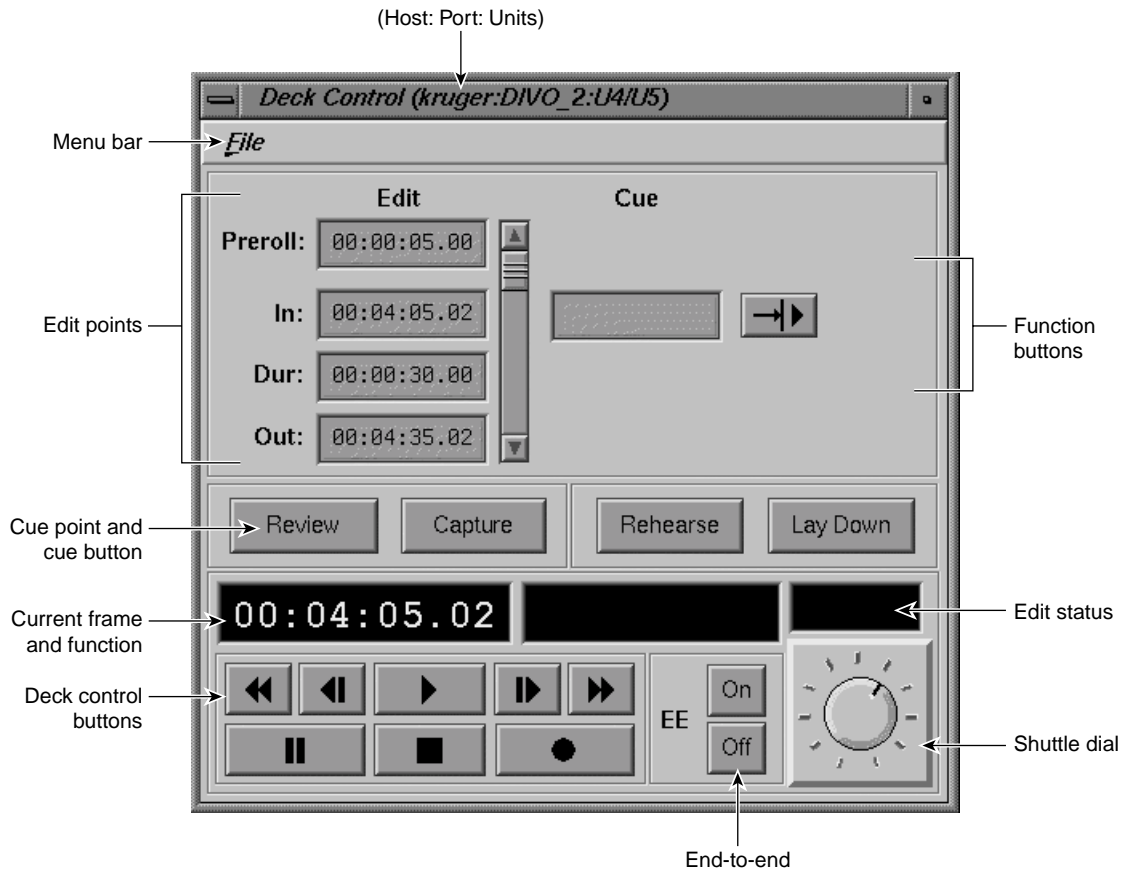
---

The following topics are discussed in this section:

- “Accessing the Deck Control Window” on page 151
- “Using the Deck Control Window” on page 152
- “Recording From the Deck to a Clip” on page 155
- “Recording From a Clip to the Deck” on page 156

## 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 A-5, appears in a separate window.



**Figure A-5** Deck Control Window

## Using the Deck Control Window

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

- |              |   |
|--------------|---|
| <b>Close</b> | Close the <b>Deck Control</b> window.   |
| <b>Exit</b>  | Close both the <b>Deck Control</b> window and the 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*  
The field *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 149 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*  
The field *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 the SGI Media Server to search the deck to the cue point and park it there.



- You can use the following deck control function buttons:

<b>Review</b>	Review the specified edit. After the edit is finished, you can use this option to examine the finished edit.
<b>Capture</b>	Perform a frame-accurate recording from the selected portion of a loaded and cued source tape to the clip currently loaded. For more information about recording, see “Recording From the Deck to a Clip” on page 155.
<b>Rehearse</b>	Practice the specified edit. The decks go through all the mechanics of a real edit except that the edit is not recorded.
<b>Lay Down</b>	Perform a frame-accurate recording from the selected portion of the clip on the SGI Media Server to the destination VTR. For more information, see “Recording From a Clip to the Deck” on page 156.
- 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:

<b>CUE</b>	System is searching for the cue point.
<b>SYNC</b>	System is doing a preroll.
<b>LOCK</b>	The deck transport is locked.
<b>EDIT</b>	The edit is in progress.
<b>DONE</b>	The edit is complete.

- The deck control buttons, which control playout and recording, correspond to standard VTR buttons. The following describes these buttons:
  - << The fast-reverse button moves the deck in reverse at a fast speed.
  - <| The jog backward button 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.
  - > The forward play button moves the deck in the forward direction. If the deck is not cued, it is cued before it begins playout.
  - |> The jog forward button 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.
  - >> The fast-forward button plays the deck forward at a fast speed.
  - | | The pause button temporarily stops the deck from playing or recording. You have to click the play, record, or pause button to resume.
  - The stop button stops the deck and de-cues it. After the deck has been stopped, you must re-cue it to play it again.
  - The record button 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

Before using the SGI Media Server to record video from a deck you must do the following:

- Connect the output of the deck to the input of the video card on the server.
- Connect the audio output of the deck to the audio inputs of the server.

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 148 for information about how to create a new clip.)
  - Change the cue in point, if needed.

---

**Note:** The cue in point in the 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 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 perform the edit, click the **Capture** button.

## Recording From a Clip to the Deck

Before using the SGI Media Server to lay back video to a deck you must do the following:

- Connect the input of the deck to the output of the video card on the server.
- Connect the audio input of the deck to the audio outputs of the server.

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 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 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.

## The Unit Status Monitor

The **Unit Status Monitor**, invoked by the `mcstat` command, graphically displays the activity of all units that are currently open on the SGI Media Server. Figure A-6 shows an example of `mcstat` output when VST is running. When VST is not running, `mcstat` exits.

Unit Status Monitor (vst12)					
MFCODEC_0					
MFCODEC_1					
MFCODEC_2	U1	nclipboard/vst12	apr9/dvctod	PLAY	16:35:37:04
MFCODEC_3	U2	nclipboard/vst12	apr9/test1	STOP	15:11:27:14 DONE
MFCODEC_4					
MFCODEC_5	U5	nclipboard/vst12	apr9/dvctod	STOP	16:44:46:09 DONE
MFCODEC_8	U4	nclipboard/vst12	apr9/I	STOP	15:15:30:11 DONE
MFCODEC_10	U3	nclipboard/vst12	apr9/test50	STOP	15:12:29:19 DONE

**Figure A-6** `mcstat` Output

The command displays a list of all the media ports supported by the SGI Media Server. For each port, it displays all the units that are opened on that port; each unit is on a separate line.

The unit display includes the name of the unit, the unit owner information, the name of the clip currently loaded on the unit, the last function executed by the unit (or the one currently executing), the current clip timecode, and the status of the function.

## The Clip Manager

This section describes how to use the Clip Manager graphical user interface (GUI) to manage media clips. Clip Manager is a sample GUI application that you can customize.

The following topics are discussed:

- “Starting the Clip Manager” on page 158
- “Clip Manager Menus” on page 161
- “Obtaining Information About a Clip” on page 162
- “Renaming a Clip” on page 163
- “Deleting a Clip” on page 163
- “Setting the Protections for a Clip” on page 164

### Starting the Clip Manager

To start the Clip Manager, enter the following command:

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

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

*hostname*           Specifies the name of the SGI Media Server, the default.

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

**Table A-3**      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

**Table A-3** Log Severity Levels (**continued**)

Priority	Log Level	Description
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 $n$	$n$	Informational message that normally is of use to engineers for debugging; may be Debug1, Debug2, or Debug3, with Debug3 producing the most debugging 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.

The Clip Manager is displayed in a separate window, which is shown in Figure A-7. The window shows the clips that are on the server. 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.)

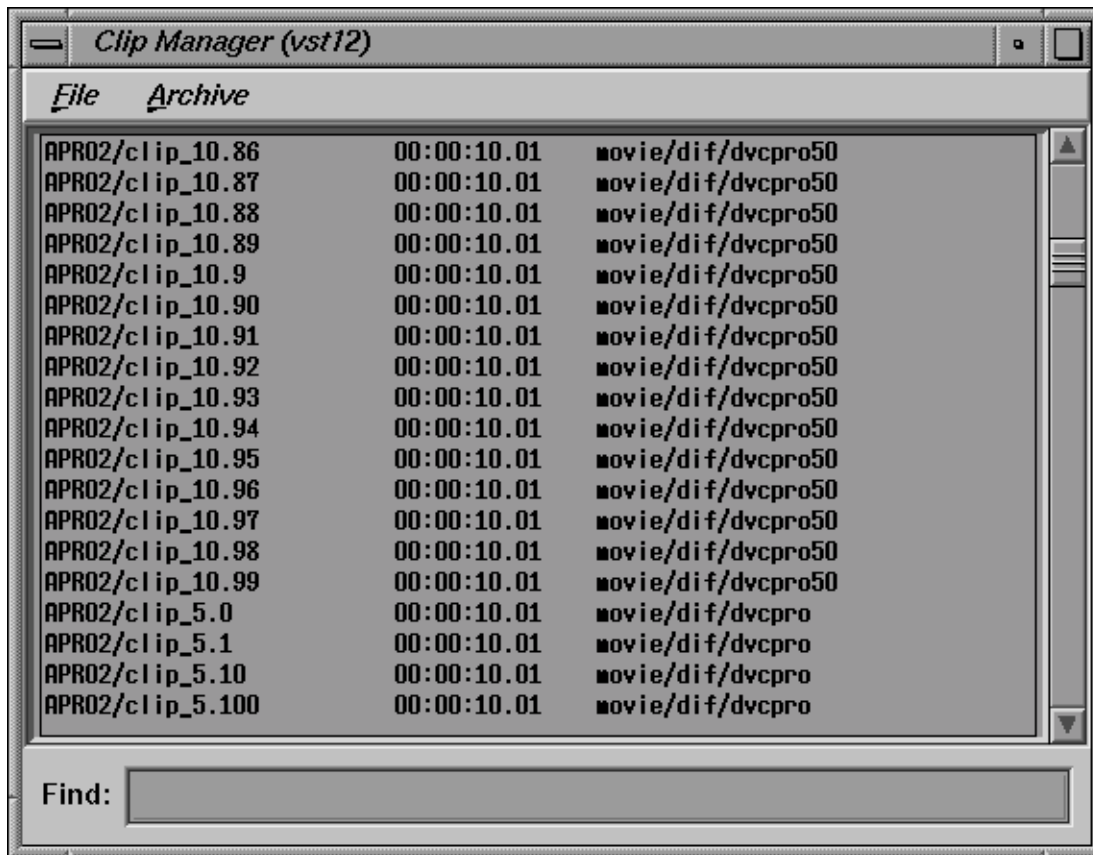


Figure A-7 Clip Manager Window

**Note:** You can also start the Clip Manager by choosing **Utilities > Clip Manager** in the **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
- **Load**, to load a clip into a unit controlled by Sony protocol

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 it finds that contains *xxx*. If you add *y* to the **Find** field, it searches for a clip that contains *xxxy*, starting with the current clip.

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

---

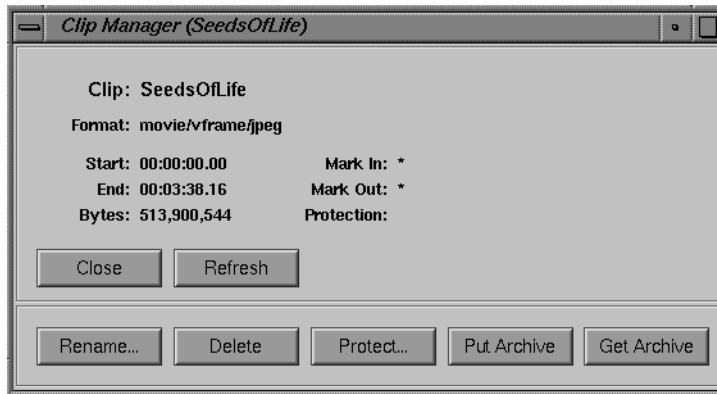
**Note:** You can access the clip-related functions in the **File** pulldown menu by selecting a clip and then pressing the right mouse button. You can also access the **Info** option by double-clicking the clip in the **Clip Manager** window.

---

## 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 A-8, appears.



**Figure A-8** 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 A-8 indicates that the edit points are not set.

The remaining sections in this chapter describe in detail the use of the buttons in this window.

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 A-9, appears.



**Figure A-9** 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 A-10, appears.



**Figure A-10** 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 SGI Media Server.

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 A-11, appears.



**Figure A-11** Set-Protections Window

This window identifies the protections that can be set. Each protection is preceded by a check box. If there is a red check in the check box, the corresponding protection is selected. If there is no 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.



---

# Glossary

## **9-Pin**

The Sony protocol for communicating with video devices. 9-Pin is the same as RS-422.

## **AES**

(Audio Engineering Society) AES Stereo Digital Audio Input/Output. This connection provides a stream of digital audio data that complies with the AES Digital Audio format.

## **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.

## **Automation Systems**

Broadcast Automation Systems (such as the Harris Automation ADC- series) are used extensively by broadcast studios to control multiple video devices. Scheduling of video playback and recording, in conjunction with pre-assigned switcher effects can all be scheduled through a single interface.

## **CCIR 601**

A digital coding standard for television that is applicable to both the NTSC as well as PAL/SECAM technologies. The standard describes the encoding parameters for the 4:2:2 member of the family (where 4:2:2 is the ratio in which the luminance and chrominance sampling frequencies are related). One CCIR 601 video stream (no alpha) is about 22 MB/sec.

## **clip**

The unit of storage in SGI Media Server. 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 the SGI Media Server 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.

**Diaquest**

Deck control software made by Diaquest.

**DIF**

The compression format used by DVCPRO. A DIF file, for example, is a file containing a clip compressed by DVCPRO.

**DSO**

Dynamic shared object. An object that one program can share with another.

**DVCPRO**

A digital video compression format provided by Panasonic. The DVCPRO 25 and DVCPRO 50 formats provide fixed media compression.

**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.”



**Fibre Channel**

The Fibre Channel Standard defines a high-speed data transfer interface that can be used to connect workstations, mainframes, supercomputers, storage devices and displays. The standard addresses the need for very fast transfers of large volumes of information. Typical high-end Fibre Channel devices carry up to 1 Gb/s data rates.

**GOP**

Group of pictures. Defined as a sequence of video frames subject to compression encoding under the MPEG-2 standard. A GOP is characterized by one of the following:

- At least one I-frame, a frame having no reference frame for prediction
- One or more P-frames, frames based on a previous frame
- B-frames, frames based on two reference frames, one previous and one afterwards

The SGI Media Server MFCODEC supports four distinct MPEG-2 GOP structures: I, IP, IPB, and IPBB.

**GOP period**

The quantity of frames in the GOP, exclusive of the I-frame. The GOP period must evenly divide the GOP size to be a valid GOP structure.

**GOP size**

The GOP size determines total number of frames in the GOP. The size can be an integer but is strictly limited by hardware/software compression algorithms and resources. An extreme upper limit for the MFCODEC GOP size is 15. Large GOPs conserve ingestion storage resources but introduce potential information loss. The GOP of a specific size and structure is encoded during video ingestion and stored by the SGI Media Server when MPEG-2 compression is selected. The GOP structure is decoded during playout.

**GRIO**

Guaranteed rate I/O. GRIO is included in IRIX and is used by the SGI Media Server to guarantee disk bandwidth for recording and playback while allowing non-priority access to the clip cache for other operations.

**Horita**

A company that makes, among other components, timecode readers. Its identifier in configuration files is hsp. See *www.horita.com*.

**hsip**

See Horita.

**Little Red**

A timecode reader made by Miranda. Its identifier in configuration files is `little-red`. See <http://www.miranda.com>.

**Material Exchange Format (MXF)**

The Material Exchange Format standardizes file structure for playable content. MXF defines operating restrictions, structural metadata, and attributes that characterize the playable content. The standard enables interoperability of playable content among disparate VTR products that can process MXF structures.

**mcpnl**

VTR emulation application for the SGI Media Server.

**media port**

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

**MFCODEC**

Multiformat codec. The compression encoding/decoder (PCI-VIDAUD-MSB-B) provided with the SGI Media Server supports three compression formats: DVCPRO 25, DVCPRO 50, and MPEG-2.

**MVCP**

Multiport Video Computer Protocol. A protocol defined by SGI that provides full-featured control of Video Server Technology through TCP/IP.

**P2**

P2 is the Panasonic name for the Sony protocol (RS-422).

**port**

A point at which an external device connects to the SGI Media Server. Video ports, used by video input and output devices. *MFCODEC* video ports are named *MFCODEC\_n* (for example, *MFCODEC\_2*). The term *media port* refers to either a video or a deck control port.

**pre-roll**

During recording, both source and recording machines can be rewound, generally three frames, so that the beginning of the recording starts when both machines are running at the correct speed. As a result, edits are more frame-accurate.

**RS-422**

Is the Sony protocol for communicating with video devices.

**SMPTE 259M**

A broadcast digital video signal standard. All high end digital video formats as well as many peripheral devices have adopted this standard. 259M is a universal standard permitting long cable runs regardless of the size of the facility.

**Sony**

A media company that uses its protocol, a standard by default, to control media machines, such as VTRs (RS-422).

**unit**

A software mechanism that functions like a logical video tape recorder transport and 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 called a *logical unit*.

**VDCP**

See Video Disk Communications Protocol.

**Video Disk Communications Protocol (VDCP)**

RS-422-based protocol used by the Harris Automation (formerly Louth) ADC series. VDCP is the basis for control protocols for other manufacturers (such as Alamar) as well. VST can parse VDCP through serial ports.

**Video Server Technology (VST)**

Video Server Technology is an element of the *Video Computing Platform (VCP)* from SGI. VST is the clip serving platform element of VCP. The core VST software provides management of a simple database of clips (the clip cache), a control API for managing and operating the VST, and a core library which supports various external interface modules.

**VTR**

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

**VST**

See Video Server Technology.

---

# Index

## Numbers

100 Base-TX network, 12, 15  
1000 Base-SX network, 12, 15  
4:2:0 compression sampling, 38  
4:2:2 compression sampling, 38  
625/50 system  
    configuring, 53  
    troubleshooting, 101

## A

ADC100 controller (See Harris Automation controllers.)  
AES/EBU audio format, 15, 26  
alert log level, 58  
analog audio format, 15, 26  
audio  
    assigning ports, 111  
    cards (See also PCI-VIDAUD-MSB-B.), 6, 12, 15  
    configuring devices, 34  
    connections, 25, 29  
    disabling, 44  
    formats, 12, 15, 26  
    identifying ports, 44  
    support, 39  
automation controllers  
    configuring, 10, 32, 45, 85, 91-96  
    general, 1, 15  
    Harris Automation, 94, 97

problems, 10  
Sony, 6, 107  
VST interfaces, 5

## B

backup, 121  
    plex, configuring boot from, 102-103  
bandwidth management, 128  
Buf RM-4000 remote processors, 15  
Buf VTC-4000 multi-VTR controller, 15  
Bufnet/RS-422 adapters, 12, 15

## C

cabling, 97  
CADD MVCP command, 64, 71, 73  
card layouts, 21, 22  
CBG. Also see IMX., 7, 12  
CBLD MVCP command, 76, 77  
CCLS MVCP command, 76, 78  
CCP MVCP command, 113  
CEDP MVCP command, 108, 113  
CFCL MVCP command, 76  
CFNW MVCP command, 76, 83  
CFRM MVCP command, 76  
CINF MVCP command, 67

- clip, 5
    - add and remove, 63
    - cache, 5, 6, 14
    - creating, 148
    - deleting, 163
    - editing, 113
    - finding a name, 108
    - loading and unloading, 108
    - managing, 158
    - mirroring, 121
    - obtaining information, 162
    - playing, 145
    - recording, 112, 145, 148
    - renaming, 163
    - setting protections, 164
    - troubleshooting, 101
  - clip alignment, 69
  - Clip Manager, 139, 158
  - CLN MVCP command, 113
  - CLS MVCP command, 108
  - CMK MVCP command, 76, 77
  - CMV MVCP command, 113
  - commands
    - sequencing, 109
  - compression
    - configuring, 36
    - DVCPRO 25, 3, 6, 7, 14, 15, 36, 65, 112
    - DVCPRO 50, 3, 6, 7, 14, 15, 36, 65, 112
    - I-frame, 14, 112
    - IMX, 6, 7, 12, 14, 36, 65, 72, 112
    - long GOP, 7, 14, 65, 112
    - MPEG-2, 3, 6, 7, 14, 15, 39, 65, 112
    - sampling, 38
  - configuration
    - audio, 39
    - card layouts, 22
    - device defaults, 34
    - external devices, 45
    - external devices, 50, 85-97
      - setting, 110
      - system defaults, 35
      - video compression, 36
  - constant-byte GOP (CBG). Also see IMX, 12
  - constant-byte GOP (CBG). Also see IMX., 7
  - control GUIs, 131
  - control interface modules, 3, 5
  - control protocols
    - MVCP, 1, 6, 8, 12, 15, 45, 85
    - Sony, 1, 12, 15, 45, 85, 86-89
    - specifying, 46
    - VDPCP, 5, 6, 12, 15, 45, 85, 91-96, 107
  - controller errors, 120
  - controllers (See automation controllers and edit controllers.)
  - COPN MVCP command, 76, 77
  - crash
    - files, 33, 103
    - semaphore limit exceeded, 102
  - critical log level, 58
  - CRM MVCP command, 72, 78, 113
  - CSAV MVCP command, 76, 78
  - CSCL MVCP command, 76, 82
  - CSLS MVCP command, 76, 81
  - CSRM MVCP command, 76, 82
  - cue
    - buttons, 140, 147
    - points, 139, 149
  - cue decks, 109
  - CUE MVCP command, 109
  - CUER MVCP command, 109
  - CUPS MVCP command, 76, 77, 78
- ## D
- D-10/IMX encoding, 15, 16, 72

DB-9 connector, 97  
 debug log level, 58  
 Deck Control Panel, 150-156  
 device  
   configuration, 34  
   external  
     configuring and using, 85-97  
     files, 34  
 device defaults, 34  
 Diaquest  
   configuring for, 34  
   controls for, 90  
   deck control module, 89  
 Diaquest software, 6, 16  
 DIF video format, 7, 14, 26, 36, 65  
 disk storage, 19  
 DMA size, setting, 53-54  
 DVCPRO 25 compression, 3, 6, 7, 14, 15, 36, 65, 112  
 DVCPRO 50 compression, 3, 6, 7, 14, 15, 36, 65, 112

## E

edit controllers, 85  
 edit decision list, 75  
 edit marks, 140, 149  
 edit point, 108  
 editing  
   clips, 113  
   frames, 114  
 embedded audio format, 15, 26  
 emergency log level, 58  
 end-to-end mode, 144, 154  
 ERR MVCP command, 120  
 error log level, 58  
 errors  
   controller, 120

MVCP, 120  
   reporting, 120  
   unit, 120  
 event logging, 57-60  
 external devices, 15  
   configuring and using, 85-97

## F

FCLR MVCP command, 114  
 Fibre Channel arrays, 12  
 file formats (See video formats.)  
 FINS MVCP command, 114  
 FNEW MVCP command, 114  
 format interface modules, 3  
 FOVR MVCP command, 114  
 frames, 83  
   editing, 114  
 FRM MVCP command, 114  
 FTP controls, 128

## G

GET MVCP command, 111  
 GOP structures, 36-39  
 gr\_osview command, 62  
 graphical user interface, 158  
 group of pictures (See also GOP.), 36

## H

hardware configuration, 22  
 hardware overview, 12  
 Harris Automation controllers, 12, 34, 91, 94, 97

## Horita

- configuring for, 97
- PR-232 timecode readers, 15, 96

**I**

- I-frame compression, 14, 37, 112
- image, displaying, emergency graphic, 114
- IMX compression, 6, 7, 12, 14, 36, 65, 72, 112
- index files, 30, 33, 69, 72, 73
- info log level, 58
- intraframe, 65
- intraframe media, 65

**L**

- LOAD MVCP command, 108
- log files, 33
- logging, 57-60
  - levels, 58
  - message structure, 58-59
  - rollover, 60
  - severity code, 59
- logo, displaying, 114
- long GOP compression, 7, 14, 65, 112
- Louth automation controllers, 171

**M**

- major alignment, 70
- Material Exchange Format (MXF) standard, 16
- mcclips, 72, 131
- mcpanel, 131
- mcstat, 131
- mcstat command, 157

## Media Control Panel, 132-156

- buttons, 141
- local mode, 144
- remote mode, 144
- starting, 133
- media device interface modules, 3, 6
- media formats (See video formats.)
- MFCODEC devices, 34, 36
- Miranda Little Red
  - configuring for, 35, 97
  - timecode reader, 15, 96
- MON MVCP command, 117
- monitor the system, 117
- monitoring VST, 61
- movie/vclip segment format, 77, 78
- MPEG-2 compression, 3, 6, 7, 14, 15, 39, 65, 112
- MSB-REARPANEL-B rear panel, 26
- multi-disk, 69
- Multiport Video Computer Protocol (See MVCP.)
- MVCP
  - overview, 1
  - protocol, 1, 6, 8, 12, 15, 45, 85
- MVCP commands
  - CADD, 64, 71, 73
  - CBLD, 76, 77
  - CCLS, 76, 78
  - CCP, 113
  - CEDP, 108, 113
  - CFCL, 76
  - CFNW, 76, 83
  - CFRM, 76
  - CINF, 67
  - CLN, 113
  - CLS, 108
  - CMK, 76, 77
  - CMV, 113
  - COPN, 76, 77
  - CRM, 72, 78, 113
  - CSAV, 76, 78



CSCL, 76, 82  
 CSLS, 76, 81  
 CSRM, 76, 82  
 CUE, 109  
 CUER, 109  
 CUPS, 76, 77, 78  
 ERR, 120  
 FCLR, 114  
 FINS, 114  
 FNEW, 114  
 FOVR, 114  
 FRM, 114  
 GET, 111  
 LOAD, 108  
 MON, 117  
 PLAY, 110  
 PLS, 111  
 REC, 112  
 SET, 32, 111  
 SGET, 111  
 SSET, 111  
 STLS, 118  
 STST, 118  
 UADD, 107  
 UCLS, 107  
 UERR, 120  
 ULS, 117  
 UNLD, 108  
 UOPN, 91, 107  
 USTA, 117  
 MVCP SET command, 32  
 MXF video format, 7, 14, 26, 36, 65, 72, 73

## N

name, of clip, 108  
 networking, 12, 15  
 non-virtual clips, 77  
 notice log level, 58

## O

OP-1a specification, MXF, 16  
 Operational Pattern 1a (OP-1a) specification, MXF, 16  
 Origin servers, 12, 14  
 osview command, 62

## P

P2 (See Sony protocol.)  
 Panasonic AG-A850 editing controllers, 12, 15  
 PCI configuration, 20  
 PCI-VIDAUD-IMX audio/video cards (See also PCI-VIDAUD-MSB-B.), 12  
 PCI-VIDAUD-MSB-B audio/video cards, 6, 12, 14, 15, 20, 26, 85  
 Performance Co-Pilot (PCP), 62  
 Perl modules, 33  
 play list, 75  
 PLAY MVCP command, 110  
 playing clips, 109  
   controlling speed, 144  
   direction, 142  
 PLS MVCP command, 111  
 ports  
   5250, 106  
   audio, 111  
   identifying, 44  
   mapping to logical port names, 52  
   media, 6, 8  
   video, 133  
 power requirements, 18  
 problems, 101  
 proxy transfers, 68

**R**

RAID storage systems, 14, 19, 69  
rear panels, 26  
REC MVCP command, 112  
recording a clip, 112  
recording video, 112  
redundancy, 121  
resource allocation, 128  
rollover, 60  
rotatelogs command, 60  
RS-422 cabling, 97  
RS-422 control protocol (See Sony control protocol.)

**S**

sar command, 62  
SCSI storage, 14  
SDI, 7, 16  
See also IMX., 15, 16  
segment  
  clearing and removing, 82  
segments, 75, 81  
sequence commands, 109  
Serial Digital Interface (SDI), 7, 16  
SET MVCP command, 111  
severity code, 59  
SGET MVCP command, 111  
SGI MPEG-2 video format, 7, 14, 26, 36, 65  
shuttle dial, 144  
site editpoints command, 67  
site marks command, 67, 72  
SMPTE 259M-1997 standard, 16  
SMPTE 266M-2002 standard, 16  
SMPTE 272M-1994 standard, 16

SMPTE 276M-1995 standard, 16  
SMPTE 314M-1999 standard, 16  
SMPTE 356M-2001 standard, 16, 36  
SMPTE-259M-1997 standard, 7  
SMPTE-314M standard, 7  
SMPTE-356M standard, 7  
SMPTE-378M standard, 7  
Sony automation controller, 107  
Sony control protocol, 1, 12, 15, 45, 85, 86-89  
space requirements, 19  
SSET MVCP command, 111  
stack trace files, 103  
standards, supported, 16  
starting  
  VST  
    automatically, 56  
    from backup plex, 102-103  
    manually, 56  
    options, 57  
statistics  
  analysis, 119  
  list, 119  
  MVCP commands, 118  
status, 141  
  of units, 117  
STLS MVCP command, 118  
stopping VST, 56  
storage, 12, 14  
storage device interface modules, 6  
STST MVCP command, 118  
system  
  defaults, configuring, 35  
  setup, 17  
  tuning for VST, 53-54

**T**

telnet, 106  
 timecode readers, 15, 96-97  
 TP900 SCSI storage systems, 12  
 troubleshooting, 101

**U**

UADD MVCP command, 107  
 UCLS command, 107  
 UERR MVCP command, 120  
 ULS MVCP command, 117  
 UltraSCSI arrays, 12  
 Unit Status Monitor, 139  
 units  
   create and delete, 107  
   errors, 120  
   logical, 8, 133  
   monitoring, 117  
   status, 117  
 UNLD MVCP command, 108  
 UOPN MVCP command, 91, 107  
 /usr/vtr/config/control-in.conf, 45  
 /usr/vtr/config/control-out.conf, 50-52  
 /usr/vtr/config/device-defaults, 34  
 /usr/vtr/config/system-defaults, 35  
 /usr/vtr/config/system-defaults/time, 35  
 /usr/vtr/config/vtrd.conf file, 52  
 USTA MVCP command, 117

**V**

VBR (variable byte rate) compression, 7  
 vclip, 75  
 VCP-Recorder Clip Manager, 72

VCP-Recorder, access manually, 106  
 VDCCP, 5, 6, 12, 15, 45, 85, 91-96, 107  
 video  
   configuration for recording, 112  
   configuring devices, 34  
   deck control, 150-156  
   ports, 133  
 video cards (See PCI-VIDAUD-MSB-B.)  
 video connections, 25  
 Video Disk Communications Protocol (See VDCCP.)  
 video formats  
   DIF, 7, 14, 26, 36, 65  
   MXF, 7, 14, 26, 36, 65, 72, 73  
   SGI MPEG-2, 7, 14, 26, 36, 65  
 Video Server Technology (See VST.)  
 virtual clips  
   file operations, 77  
   frames, 83  
   overview, 75  
   segments, 78  
 VST  
   checking status, 57  
   events, 57-60  
   monitoring, 61  
   overview, 3, 5  
   starting  
     automatically, 56  
     manually, 56  
     options, 57  
   stopping, 56  
 VTR RS-422 control protocol (See Sony control protocol.)  
 VTR, configuring VST to control, 89-91  
 vtrclip, 71, 72  
 vtrclip command, 73  
 vtrclip tool, 64  
 vtr.control.clip.name, 88, 89  
 vtr.control.device\_type\_id, 88

- vtr.control.edit.delay, 88
- vtr.control.ee.delay, 88
- vtr.control.ee.mode, 88
- vtr.control.ee.record\_select, 88
- vtr.control.output.idle\_mode, 88
- vtr.control.superimpose.enabled, 88
- vtr.control.timecode.mode, 88
- vtrd process, 52
- vtrd server daemon, 32
- vtrdircopy daemon, 32
- vtr.edit.coincidence.preroll, 91
- vtr.edit.postroll, 88, 90
- vtr.edit.preroll, 88, 90
- vtrftp command, 68
- vtrftpd daemon, 66, 67
- vtrhwinfo command, 44
- vtr.main.log\_level.file, 59
- vtr.main.timing\_standard, 35, 53, 101
- vtr.media.audio.input.channel\_map.channel\_pair, 34, 42, 44
- vtr.media.audio.input.channel\_map.source, 42, 44
- vtr.media.audio.input.channels, 34, 44
- vtr.media.audio.output.port, 34
- vtr.media.clip.start.mode, 34

- vtr.media.mpeg.bit\_rate, 34, 37
- vtr.media.output.mode, 91
- vtr.media.sync\_port, 91
- vtr.media.video.input.compression.gop\_size, 38
- vtr.media.video.input.compression.gop\_structure, 38
- vtr.media.video.input.compression.sampling, 38
- vtr.media.video.input.compression.type, 34, 36, 38
- vtr.media.video.input.vitc.line\_offset, 34
- vtr.media.video.output.timing, 53, 101
- vtrmpegutil command, 73
- vtrstat, 61
- vtr.time.offset, 35
- vvtr process, 52

## W

- warning log level, 58

## X

- XFS filesystem, 128
- XFS real-time filesystem, 66