# VCP-Recorder
# Installation and System Administration Guide

Document Number 007-3622-001

CONTRIBUTORS

Written by George Eckel
Illustrated by Dany Galgani
Production by Allen Clardy
Engineering contributions by Rick Reed and Dave Worthington.

VCP-Recorder Installation and System Administration Guide
Document Number 007-3622-001

# Contents

# List of Examples

# List of Figures

# List of Tables

# About This Guide

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

VCP-Recorder plays and records digital media data and stores the data in, and retrieves the data from, the MediaHub archive system. For more information about using VCP-Recorder, see the *VCP-Recorder User's Guide*.

The *VCP-Recorder Installation and System Administrator's Guide* describes how to install and manage VCP-Recorder in the following sections:

- "What This Document Contains" on page xiii.
- "Who Should Read This Document" on page xiv.
- "Related Documentation" on page xiv.
- "Conventions Used in This Document" on page xvi.

## What This Document Contains

The following material is covered in this document:

### Part I, "Installing and Configuring VCP-Recorder"

- Chapter 1, "Installing VCP-Recorder," provides a step-by-step description of installing the VCP-Recorder software.
- Chapter 2, "Installing the Louth Automation Controller," describes how to configure the Louth automation controller to work with VCP-Recorder.
- Chapter 3, "Installing the Vela Research MPEG-2 Decoder," describes how to install and configure the Vela decoder.
- Chapter 4, "Installing V-LAN Transmitters," describes how to integrate V-LAN and VCP-Recorder.

### Part II, “Managing VCP-Recorder”

- Chapter 5, “Starting and Stopping VCP-Recorder,” explains how to start VCP-Recorder.
- Chapter 6, “Logging,” describes the diagnostic log messages output by VCP-Recorder.
- Chapter 7, “Managing Clips,” describes how to add and remove audio and video media clips from VCP-Recorder.
- Chapter 8, “Monitoring the System,” describes how to monitor VCP-Recorder.

## Who Should Read This Document

This document is written for VCP-Recorder system administrators responsible for installing and monitoring the running of VCP-Recorder.

## Related Documentation

Refer to the following documents for related information:

- *VCP-Recorder User’s Guide*, for information about using VCP-Recorder to play and record digital media data; and store the data in, and retrieve it from, the MediaHub *archive system*.
- *MediaHub User’s Guide*, for information about using MediaHub, including the MediaHub Console application and the MediaHub command-line utilities
- *MediaHub Programmer’s Guide*, for information about using the application programming interface for the MediaHub archive system

Refer to the following documents for supplementary information:

- *IRIX Admin: Software Installation and Licensing* (part number 007-1364-*nnn*) for information about installing software that runs under IRIX, the Silicon Graphics implementation of the UNIX operating system

- *IRIX Admin: System Configuration and Operation* (part number 007-2859-*nnn*) for information about IRIX system administration tasks
- *IRIX Admin: Disks and Filesystems* (part number 007-2825-*nnn*) for information about general filesystem concepts and system administration procedures for SCSI disks, XFS and EFS filesystems, logical volumes, and guaranteed rate I/O

## Man Pages

VCP-Recorder contains the following man pages:

- mcclips
- mccompstats
- mcpanel
- mcstat
- mvcp
- vcp-recorder-controls
- vcp-recorder
- vtrclip
- vtrd
- vtrstart
- vtrstat
- vtrstop
- vtrvfutil
- vvtr

You can list the man pages by entering the following command:

```
% versions long vcp_recorder_eoe | grep man
```

## Conventions Used in This Document

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

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

**Bold** used for keywords.

`Fixed-width` used for code examples and command syntax.

**`Bold fixed-width`**
used for user input, including nonprinting keyboard keys.

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

Square bullets (■)
indicate substeps within a multi-step process.

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

Right angle brackets (>)
indicate a path through menus to a menu option. For example, “File > Open” means “Under the File menu, choose the Open option.”

PART ONE

# Installing and Configuring VCP-Recorder

This part of the book explains how to install and configure VCP-Recorder and several external devices that work in concert with VCP-Recorder.

Chapter 1, “Installing VCP-Recorder.”

Chapter 2, “Installing the Louth Automation Controller.”

Chapter 3, “Installing the Vela Research MPEG-2 Decoder.”

Chapter 4, “Installing V-LAN Transmitters.”

*Chapter 1*

# Installing VCP-Recorder

This chapter provides a step-by-step description of installing the VCP-Recorder software in the following sections:

- "Supported Hardware" on page 3.
- "Pre-Installation Steps" on page 5.
- "Installing VCP-Recorder" on page 7.
- "Configuring VCP-Recorder" on page 13
- "Confirming the Installation" on page 17.

## Supported Hardware

VCP-Recorder works with the following servers and workstation:

- O2
- Origin200
- Origin2000
- Onyx2

## External Devices

VCP-Recorder can interact with a variety of external devices, as shown in Figure 1-1.

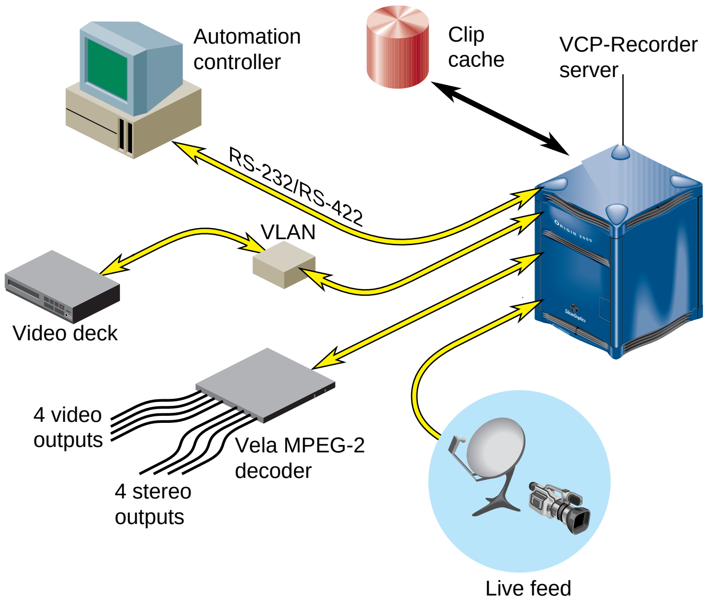


**Figure 1-1** VCP-Recorder with External Devices

Part of the installation of VCP-Recorder includes configuring it to work with these input devices.

VCP-Recorder works with the following external devices:

- Videotape recorders; V-LAN Transmitter; see Chapter 4, “Installing V-LAN Transmitters.”
- Louth Automation Controller; see Chapter 2, “Installing the Louth Automation Controller.”
- Vela Research MPEG-2 Decoder; see Chapter 3, “Installing the Vela Research MPEG-2 Decoder.”

## Pre-Installation Steps

Before you install VCP-Recorder, read through the following pre-installation topics to make sure you have the materials required for the installation and that your workstation meets all the VCP-Recorder requirements:

- “Collecting Required CDs” on page 5.
- “Reading the Release Notes” on page 5.
- “Installing Patches” on page 6.
- “Installing Additional Hardware” on page 6.
- “Verifying Your Network Connection” on page 7.
- “Using IRIX 6.3 or Later” on page 7.

### Collecting Required CDs

To install VCP-Recorder, you need the following CDs:

- VCP-Recorder CD
- IRIX CD
- DIVO CD, if using DIVO video board(s)
- Patches CD for any patches required for the operation of VCP-Recorder not already included on the distribution CD.

  To determine which patches are required for the successful installation of VCP-Recorder, see the Release Notes.

### Reading the Release Notes

The release notes contain the latest information for the software required to use VCP-Recorder, such as required patches.

To read the release notes from the CD, use the following command:

```
# /CDROM/doc6.x/CDgrelnotes
```

*dic6.X* is either **doc6.3** or **doc6.4**, depending on the operating system your machine is running.

## Installing Patches

Before installing VCP-Recorder, install the patches required for VCP-Recorder, as specified in the Release Notes.

Use Software Manager or the *inst* program to install any required patches.

## Installing Additional Hardware

Before installing VCP-Recorder, install any hardware related to VCP-Recorder, such as VLAN, Vela, or DIVO boards. When you install VCP-Recorder, it automatically looks for drivers for optional hardware and changes its own default installation to include the subsystems relevant to the optional hardware.

For information about installing the optional hardware, please refer to the information accompanying the hardware.

### Video Hardware

The video hardware is an optional upgrade to your machine, but required for using VCP-Recorder.

An O2 needs one of the following video boards: Video I/O (also known as AV1, or O2Video) or Digital Video I/O (also known as AV2, or O2Video-601).

To determine whether your O2 has a video board, see if your machine has six RCA sockets: two red, two while, and two yellow. If you cannot find these sockets on your machine, you most likely do not have a video board.

An Origin2000, or Onyx2 needs a DIVO (Digital Video Option) board.

**Note:** VCP-Recorder is not supported using DIVO on an Origin200.

For MPEG-2 decoding, you need a Vela 4-port, SCSI-attached, MPEG-2 decoder.

### Verifying Your Network Connection

To install VCP-Recorder over a network, you must be successfully connected to a network. The default installation of an O2, for example, does not include networking software. To connect to a network, you must purchase NFS separately and install it.

Once NFS is installed, you create the file, */var/yp/ypdomain* and only include in it the domain name of your machine.

For more information about connecting to a network, consult your system administrator.

### Using IRIX 6.3 or Later

VCP-Recorder only works with IRIX 6.3 or later. This requirement means, for example, that VCP-Recorder does not work on a Challenge.

## Installing VCP-Recorder

Once your machine is properly configured, use the following procedure for installing VCP-Recorder over a network or from a CD.

1. Become a super user by entering the following command:

   ```
   % su
   ```

2. Start the installation program and identify the source of the VCP-Recorder image, as follows:

   ```
   # inst -f <machineName>:<distribution-directory>
   ```

   Or, if you are installing from a CD, use

   ```
   # cd /CDROM
   # inst -f dist
   ```

   This installation works for all operating systems.

   If you are installing over the network, use a line similar to the following:

   ```
   # inst -f bubka:dist/vcp-recorder/dist
   ```

3. List the subsystems to install as follows:

```
Inst> l
  View:      distribution
  Status:    N=new, U=upgrade, S=same, D=downgrade, P=patch upgrade
  Patches:   A=installable patch, X=uninstallable (missing base
product)
  Selection: i=install, r=remove, k=keep

  Subsystem Types [bdrpc]:  b=reBoot needed, d=Default, r=Required,
p=patch, c=Client only

  S  Das_lite_eoe.sw.lib [d]            0   MediaHub/MediaCache
Interface Library

  N  fw_LWperl5.man.mansrc           1984+  Perl5 man page source
  S  fw_LWperl5.man.pages [d]           0   Perl5 formatted man
pages
  S  fw_LWperl5.man.relnotes [d]        0   Perl5 release notes
  N  fw_LWperl5.src.perl             7204+  Base Perl Source
  S  fw_LWperl5.sw.perl [d]             0   Perl5 interpreter and
std library

  S  fw_common.man.legal [d]            0   Freeware Legal Notice
  S  fw_common.sw.fixpath [d]           0   Freeware Common Tools
  S  fw_common.sw.index [d]             0   Freeware Release Note
Index Builder

  PA patchSG0002275.desktop_base_sw.FileTypingRules [p]  0
Desktop Filetype Rules
  NX patchSG0002275.dmedia_dev_man.pages [p]  20+  Digital Media
Development Man Pages
  NX patchSG0002275.dmedia_dev_sw.base [p]  180+  Digital Media
Development Environment
  NX patchSG0002275.dmedia_dev_sw.debug [p]  4556+  Digital Media
Debug Libraries
  NX patchSG0002275.dmedia_dev_sw32.debug [p]  4020+  Digital Media
Debug n32 Libraries
  PA patchSG0002275.dmedia_eoe_man.pages [p]  0   Digital Media
Manual Pages
  PA patchSG0002275.dmedia_eoe_sw.base [bp]  0   Digital Media
System Components
  PA patchSG0002275.dmedia_eoe_sw.lib [p]  4+  Digital Media
Execution Libraries
```

```
  PA patchSG0002275.dmedia_eoe_sw.plugins [p]  0   Digital Media
Plug-ins
  PA patchSG0002275.dmedia_eoe_sw.tools [p]  0   Digital Media Tools
  PA patchSG0002275.dmedia_eoe_sw32.lib [p]  0   Digital Media n32
Execution Libraries
  PA patchSG0002275.dmedia_eoe_sw32.plugins [p]  0   Digital Media
n32 Plug-ins
  PA patchSG0002275.media_warehouse_sw.viewers [p]  0
MediaWarehouse Viewer Framework

  NX patchSG0002485.dmedia_dev_man.pages [p]  0   Digital Media
Development Man Pages
  NX patchSG0002485.dmedia_dev_sw.base [p]  60+  Digital Media
Development Environment
  NX patchSG0002485.dmedia_dev_sw.debug [p]  536+  Digital Media
Debug Libraries
  NX patchSG0002485.dmedia_dev_sw32.debug [p]  236+  Digital Media
Debug n32 Libraries
  SA patchSG0002485.dmedia_eoe_man.pages [p]  0   Digital Media
Manual Pages
  SA patchSG0002485.dmedia_eoe_sw.base [p]  0   Digital Media
System Components
  SA patchSG0002485.dmedia_eoe_sw.lib [p]  0   Digital Media
Execution Libraries
  SA patchSG0002485.dmedia_eoe_sw32.lib [p]  0   Digital Media n32
Execution Libraries

  NX patchSG0002486.dmedia_dev_man.pages [p]  400+  Digital Media
Development Man Pages
  NX patchSG0002486.dmedia_dev_src.examples [p]  1580+  Digital
Media Example Source Code
  NX patchSG0002486.dmedia_dev_sw.base [p]  24+  Digital Media
Development Environment
  SA patchSG0002486.dmedia_eoe_books.MediaTls_UG [p]  0   Media
Tools User’s Guide, Help and Book
  NX patchSG0002486.dmedia_eoe_data.synth [p]  772+  MIDI
Synthesizer Data Files
  SA patchSG0002486.dmedia_eoe_man.pages [p]  0   Digital Media
Manual Pages
  SA patchSG0002486.dmedia_eoe_man.relnotes [p]  0   Digital Media
Release Notes
  SA patchSG0002486.dmedia_eoe_sw.audio [bp]  0   Audio System
Components
  SA patchSG0002486.dmedia_eoe_sw.base [p]  0   Digital Media
System Components
```

```
  SA patchSG0002486.dmedia_eoe_sw.lib [p]  0   Digital Media
Execution Libraries
  NX patchSG0002486.dmedia_eoe_sw.synth [p]  1236+  MIDI Synthesizer
  SA patchSG0002486.dmedia_eoe_sw.tools [p]  0   Digital Media Tools
  SA patchSG0002486.dmedia_eoe_sw32.lib [p]  0   Digital Media n32
Execution Libraries
  SA patchSG0002486.eoe_sw.unix [p]     0   IRIX Execution
Environment

  PA patchSG0002796.eoe_sw.unix [bp]    0   IRIX Execution
Environment

  S  vcp_recorder_eoe.books.AdminGuideHTML  0   VCP-Recorder 1.0
HTML Administrators Guide
  S  vcp_recorder_eoe.books.UsersGuideHTML  0   VCP-Recorder 1.0
HTML Users Guide
  S  vcp_recorder_eoe.books.VCPRecorderAG  0   VCP-Recorder 1.0
Administrators Guide
  S  vcp_recorder_eoe.books.VCPRecorderUG  0   VCP-Recorder 1.0
Users Guide
  S  vcp_recorder_eoe.man.base [d]      0   VCP-Recorder 1.0 Base
man pages
  S  vcp_recorder_eoe.man.relnotes [d]  0   VCP-Recorder 1.0
Release notes
  S  vcp_recorder_eoe.man.tools [d]     0   VCP-Recorder 1.0 UI
applications man pages
  S  vcp_recorder_eoe.sw.base [d]       0   VCP-Recorder 1.0 base
software
  N  vcp_recorder_eoe.sw.louth        188+  VCP-Recorder 1.0 Louth
processor
  S  vcp_recorder_eoe.sw.mediahub       0   VCP-Recorder 1.0
MediaHub interface
  N  vcp_recorder_eoe.sw.mpeg         200+  VCP-Recorder 1.0 MPEG
tools
  S  vcp_recorder_eoe.sw.tools [d]      0   VCP-Recorder 1.0 UI
applications
  N  vcp_recorder_eoe.sw.vela         168+  VCP-Recorder 1.0 Vela
SCSI MPEG-decoder interface
  S  vcp_recorder_eoe.sw.vl             0   VCP-Recorder 1.0 SGI
DIVO/O2Video interface
  N  vcp_recorder_eoe.sw.vlan         120+  VCP-Recorder 1.0 VLAN
deck-control interface

Disk space summary (Kbytes):            / /usr/vtr/clips
```

```
Current free space                963716         185156
- Selections net change                0              0
- Temporary inst overhead              0              0
= Minimum free during install     963716         185156

Final projected free space        963716         185156
```

**Note:** If you installed the drivers for hardware, such as a VELA or VLAN board, the subsystem for those pieces of hardware would be marked for installation by default.

This listing is for IRIX 6.3. The listing is slightly different for IRIX 6.4. Your operating system is read automatically and the correct subsystems are listed for your machine.

Table 1-1 describes each of the subsystems.

**Table 1-1** VCP-Recorder Subsystems

| If Using | Required Subsystems | Description |
|---|---|---|
| VCP-Recorder | vcp_recorder_eoe.sw.base | VCP-Recorder Base software. |
| O2 / O2Video or Origin / DIVO | vcp_recorder_eoe.sw.vl | VCP-Recorder SGI DIVO / O2Video interface. |
| MPEG decoding using Vela decoder (O2 or Origin) | vcp_recorder_eoe.sw.mpeg | VCP-Recorder MPEG tools. |
| | vcp_recorder_eoe.sw.vela | VCP-Recorder Vela SCSI MPEG-decoder interface. |
| VLAN transmitter | vcp_recorder_eoe.sw.vlan | VCP-Recorder VLAN deck-control interface |
| Automation controller using Louth protocol | vcp_recorder_eoe.sw.louth | VCP-Recorder Louth processor |
| MediaHub archive system | vcp_recorder_eoe.sw.mediahub | VCP-Recorder MediaHub interface. |
| Operator tools | vcp_recorder_eoe.sw.tools | VCP-Recorder UI applications. |
| Documentation | vcp_recorder_eoe.man.tools | VCP-Recorder UI applications man pages. |
| | vcp_recorder_eoe.man.base | VCP-Recorder Base man pages. |

**Table 1-1** VCP-Recorder Subsystems

| If Using | Required Subsystems | Description |
|---|---|---|
| | vcp_recorder_eoe.man.relnotes | VCP-Recorder Release notes |
| | vcp_recorder_eoe.books.* | Two books, each in html and Insight:<br>* *VCP-Recorder Installation and System Administration Guide.*<br>* *VCP-Recorder User's Guide.* |

**Note:** The documentation in *vcp_recorder_eoe.books.** is not installed by default.

4. Mark for installation any optional VCP-Recorder-related software you might need. For example, if you are using VELA, MPEG, louth, or VLAN, install one or more of the following options, respectively:

```
Inst> i vcp_recorder_eoe.sw.vela
Inst> i vcp_recorder_eoe.sw.mpeg
Inst> i vcp_recorder_eoe.sw.louth
Inst> i vcp_recorder_eoe.sw.vl
```

For more information about VELA, see Chapter 3, "Installing the Vela Research MPEG-2 Decoder."

For more information about louth, see Chapter 2, "Installing the Louth Automation Controller."

**Tip:** If you intend to install MediaHub (*vcp_recorder_eoe.sw.mediahub*) for clip archiving, you need to install *Das_lite_eoe.sw* from the MediaHub CD first.

5. Initiate the installation, as follows:

```
Inst> go
```

6. Quit the installation program, as follows:

```
Inst> quit
```

## Installing a Different Version of VCP-Recorder

VCP-Recorder automatically detects the operating system of your machine and lists the subsystems for that operating system automatically. If you would like to install the 6.4 version of VCP-Recorder on a machine running IRIX 6.3, use the following installation command in step 2 above:

```
# inst -f dist6.4
```

You can also install the 6.3 version of VCP-Recorder on a machine running IRIX 6.4, use the following installation command in step 2 above:

```
# inst -f dist6.3
```

## Configuring VCP-Recorder

Use the following procedure to configure VCP-Recorder:

1. Move to the configuration file directory, as follows:

   ```
   # cd /usr/vtr/config
   ```

2. If you are connecting to MediaHub for archiving clips, using a text editor, append the following line to the configuration file, *mediahub.conf*, if Mediahub is running on a remote machine:

   ```
   MediahubServerMachineName    ats
   ```

   *MediahubServerMachineName* is the host name of the server running Mediahub. You can add more than one server name. The MediaHub Asset Transfer Services (ATS) should always be specified as the second argument. This argument starts ATS on the server.

   If Mediahub is running on your local machine, append “share” to the line, for example:

   ```
   myMachine    ats    share
   ```

To configure VLAN or Louth, see Chapter 4, “Installing V-LAN Transmitters” and Chapter 2, “Installing the Louth Automation Controller,” respectively.Configuring Automatic Startup at Boot Time

By default, when VCP-Recorder is installed, it does not enable itself to be automatically started when the system boots.   VCP-Recorder can be configured to automatically start itself at boot time using chkconfig. VCP-Recorder is controlled by the setting of the vtr option:

```
% chkconfig vtr on      # enables automatic startup
% chkconfig vtr off     # disables automatic startup
```

## Configuring the Clip Cache

All clips played and recorded by VCP-Recorder are stored on one or more XFS-based file systems referred to as the “clip cache.” These file systems are mounted at or under the directory */usr/vtr/clips*.

### Creating the Clip Cache File System

How the file system which will host the clip cache is created depends upon the type of disk subsystem that will be used. Available choices for clip storage include a native XFS file system on an attached disk drive, an XLV striped disk volume (using UltraSCSI disk drives on internal and / or external UltraSCSI buses), or some type of RAID subsystem connected via either FibreChannel or SCSI. If an XLV striped disk volume is to be used, *xlv_make*(1M) is used to create the logical volume. The volume’s stripe unit (the number of disk blocks per disk per stripe) is chosen to maximize performance.

A stripe unit value of 552 disk blocks is optimal for 525-line uncompressed (Rec. 601) video use. This results in a stripe size of 2260992 bytes for an 8-disk volume, or 1081344 bytes for a 4-disk volume. This stripe size holds, respectively, exactly 6 or 3 uncompressed 4:2:2 8-bit fields (2 bytes / pixel) when the field is rounded up to the nearest 16KB to match the recommended file system block size.

A stripe unit value of 624 disk blocks is optimal for 625-line uncompressed (Rec. 601) video use. This results in a stripe size of 2654208, bytes for an 8-disk volume, or 1327104 bytes for a 4-disk volume. This stripe size holds, respectively, exactly 6 or 3 uncompressed 4:2:2 8-bit fields (2 bytes / pixel) when the field is rounded up to the nearest 16KB to match the recommended file system block size.

For compressed video, a stripe unit of 512 disk blocks should work well.

Once the XLV volume is created, or if you are using an external RAID device, you must create the XFS file system using *mkfs_xfs*. A file system block size of 16KB is suggested:

```
% mkfs_xfs -b size=16384 <disk volume device>
```

**Note:** If you wish to use a real-time subvolume for the clip cache, install the latest recommended XFS rollup patch for your OS version.

In addition, you must set the real-time extent size equal to the stripe size of your disk subvolume.

### Mounting the Clip Cache File System(s)

If the clip cache will occupy only a single file system, it can be mounted as */usr/vtr/clips*. If, however, the clip cache will extend across several file systems, those file systems must be mounted under */usr/vtr/clips* (for example, as */usr/vtr/clips/a* and */usr/vtr/clips/b*).

## Configuring the Video Interface

Default control settings for your video devices may be set by creating or editing the appropriate configuration file in */usr/vtr/config/device-defaults*. To make control settings that are used by all DIVO video ports, you may set the default values that are used when each unit is created by editing */usr/vtr/config/device-defaults/DIVO*.

Default control values may be set for individual DIVO board by creating the appropriate device-defaults file using the name of the DIVO port. For example, device-default control values that apply only to the first DIVO port are placed in:
*/usr/vtr/config/device-defaults/DIVO_0*.

Refer to the *vcp-recorder-controls(5)* man page for detailed information about the configuration controls available.

## Configuring Audio Support

**Note:** This section does not apply to audio handling when a Vela MPEG decoder is being used. The audio configuration is managed by the decoder.

The default input audio port is “DefaultIn,” meaning the audio is taken from the default input as set in the server’s Audio Panel. Similarly, the default output audio port is “DefaultOut”.

To change the input or output audio port, set either vtr.media.audio.input.port or vtr.media.audio.output.port, respectively. Audio port names take the form subsystem.interface.jack

For audio ports based upon the Radical audio subsystem:

- *subsystem* is “RADn.”
- *interfaces* is one of “AnalogIn,” “AnalogOut,” “AESIn,” “AESOut,” “ADATIn,” or “ADATOut.”

- *jack* is one of “LineIn,” “Microphone,” and only applies to “AnalogIn.”

For example, to take audio input for the DIVO_1 port from the AES coaxial input on a Graphics BaseIO, edit the file */usr/vtr/config/device-defaults/DIVO_1*, and add this control setting:

```
vtr.media.audio.input. port “RAD1.AESIn”
```

To use the SMPTE-272M embedded audio support on a DIVO port:

- *subsystem* is “DIVO_n”
- *interface* is “DigitalIn” or “DigitalOut”

You may also specify just “VideoIn” or “VideoOut” (no subsystem name). These interface names are a special tokens that VCP-Recorder translates to the appropriate DIVO digital audio port, for example:

```
vtr.media.audio.output.port “VideoOut”
```

## Setting Video Output Sync Source

The *vtr.media.video.output.sync_source* control may be set to one of internal, external, *digital-input-link-a*, or *digital-input-link-b*.

This control may be set for all DIVO boards in the server in the *device-defaults/DIVO* file, or may be set per board in the *device-defaults/DIVO_n* files.

## Setting the Compression Type

By default, VCP-Recorder uses Rice packing when recording clips using a DIVO and JPEG compression when recording clips on an $O_2$.

To capture uncompressed video fields, set the *vtr.media.video.input.compression.type* control to none.

## Configuring the MediaHub Interface

To enable VCP-Recorder to use one or more MediaHub hosts as archive systems, edit */usr/vtr/config/mediahub.conf*, and include one line for each MediaHub host.

The format of each configuration line is:

```
<hostname> <service-name>
```

Usually, the *service-name* is “ats”.

### Sharing Media Assets Between MediaHub and VCP-Recorder

If the MediaHub host is the same as the VCP-Recorder host, the media files can be shared by appending “shared-content” to the end of the configuration line, as follows:

```
machineName    ats     shared-conten
```

where *machineName* is the VCP-Recorder and MediaHub host. Sharing the assets prevents duplication of the media assets on your server.

## Confirming the Installation

To confirm the successful installation of VCP-Recorder, complete the following steps:

1. Start VCP-Recorder by entering the following command as a super user:

   ```
   # /usr/vtr/bin/vtrstart
   ```

   If VCP-Recorder starts, on an O2 only, the VCP-Recorder window displays.

2. To see if VCP-Recorder started, run

   ```
   # /usr/vtr/bin/vtrstat
   ```

   If VCP-Recorder is running, the command responds:

   ```
   VCP-Recorder on hostname is running.Verify the media ports supported
   by running the following command:
   ```

   ```
   /usr/vtr/bin/vtrstat -ports
   ```

   vtrstat lists the supported ports, for example:

   ```
   #  Port    Type    Description
   ----------------------------------------------------------
   0  vlan_0  Deck    VLAN Deck Control
   1  DIVO_0  Video   SGI XT-DIVO Digital Video Option
   2  DIVO_1  Video   SGI XT-DIVO Digital Video Option
   3  DIVO_3  Video   SGI XT-DIVO Digital Video Option
   4  DIVO_4  Video   SGI XT-DIVO Digital Video Option
   5  DIVO_5  Video   SGI XT-DIVO Digital Video Option
   ```

```
6  DIVO_6  Video   SGI XT-DIVO Digital Video Option
7  DIVO_7  Video   SGI XT-DIVO Digital Video Option
```

3. You can also confirm the installation by inspecting the contents of the /*usr/vtr* and */var/adm/vtr/logs* directories:

```
dutch 2# cd /usr/vtr
dutch 3# ls
bin      clips    config   data     images   index    install
lib32    lib
dutch 4# cd /var/adm/vtr
crash   logs
dutch 5# cd /var/adm/vtr/logs
dutch 6# ls
vtrlog
```

The crash directory contains crash files for *vvtr* and *vtrd*.

Table 1-2 describes these directories.:

**Table 1-2** /usr/vtr Subdirectories

| Subdirectory | Description |
|---|---|
| bin/* | Commands and VCP-Recorder executable, *vvtr*. |
| config/* | Configuration files. |
| index/* | Clip indices (MPEG and vframe formats). |
| lib32/*.so | Core and shared libraries. |
| lib32/modules/*.so | External interface device/control/format modules. |
| clips/* | Clip cache (can be symlink). |
| images/* | The directory under which still images are created when the MVCP CIMG command is executed. |
| data/* | Files under this directory are static data or image files used by various VCP-Recorder components. |
| install/ | Files that are used during the installation. |

## Common DSOs

The following common DSOs are installed in */usr/vtr/lib32*:

**Table 1-3** Common DSOs

| Common DSOs | Description |
|---|---|
| libVtr.so | VVTR core library. |
| libVtrControlCommon.so | Common controller code. |
| libVtrMvcpClient.so | Common MVCP code. |

*Chapter 2*

# Installing the Louth Automation Controller

The Louth Video Disk Communications Protocol, which is defined by Louth Automation, provides full-featured control of the VCP-Recorder using RS-232, RS-422, and TCP/IP. The VCP-Recorder's Louth processor supports back-to-back play and record (subject to restrictions imposed by the video I/O port capabilities) and archival management.

This chapter provides a step-by-step description of installing Louth in the following sections:

- "Louth Video Disk Communications Protocol" on page 21.
- "Connecting the Louth Automation Controller to VCP-Recorder" on page 22.

## Louth Video Disk Communications Protocol

At the time of this writing, all Louth Video Disk Communications Protocol commands defined as of the 2/22/95 protocol revision are supported except for the following:

```
Local Disable
Local Enable
Video Compression Rate
Audio Sample Rate
Audio Compression Rate
Audio IN Level
Audio OUT Level
Video Compression Parameters
Select Output
Select Input
Record Mode
SC Adjust
Horizontal Position Adjust
Compression Settings Request
```

## Connecting the Louth Automation Controller to VCP-Recorder

An automation controller is connected to a VCP-Recorder server using one or two serial ports per video port to be controlled. If the VCP-Recorder server supports simultaneous play and recording, two serial port connections are required if you want to be able to play and record at the same time.

Origin / DIVO and O2 / O2Video support simultaneous playing and recording. Vela is playback only.

O2 serial ports run only in RS-232 mode. Consequently, to connect the Louth ADC-100 automation controller to the serial ports of the O2, you do one of the following:

- Insert a 422 / 232 protocol converter between the Louth and the O2.
- Use a special cable that uses the A (negative) and ground line from the RS-422 to provide a RS-232-compatible connection.

The Origin / Onyx2 serial ports can run in either RS-232 or RS-422 mode. The configuration file (louth.conf) specifies which mode to use.

See the serial(7) man page for information about the RS-232 and RS-422 ports.

### Configuring the Louth ADC-100 Automation Controller

There are two steps to configure the Louth ADC-100 automation controller:

1. “Configure the Louth ADC-100” on page 23
2. “Configure the Ports on the VCP-Recorder System” on page 25

### Configure the Louth ADC-100

On the Louth ADC-100, configure the VCP-Recorder system as a standard video disk. Table 2-1 shows how to specify the device parameters for the video disk.,

**Table 2-1** Louth Device Parameters

| Device Parameter | Setting |
|---|---|
| AUDIO / VIDEO SWITCHING DATA | All zero |
| DEVICE NAME | User-specified |
| VIDEO INPUT PORT IN DISK | Video port to use for recording |
| VIDEO OUTPUT PORT IN DISK | Video port to use for playing |
| REPORT DISK SPECIFIC WARNINGS | Enabled |
| STOP DISK PLAY / RECORD ON INIT | Enabled |
| ENABLE SEGMENT PLAY / RECORD WITHIN ID | Enabled |
| UPDATE EVENT DURATIONS FROM DISK | Enabled |
| CONFIGURE INSTANT PLAY PREROLLS | Enabled, 0 seconds 4 frames |
| CONFIGURE DISK PREROLLS | 3 frames |
| NUMBER (=) OF FRAMES TO SEND PLAY EARLY | 3 frames |
| ENABLE BACK TO BACK PLAY | Enabled |
| DISK PORT IS AIR PROTECTED | As appropriate |
| MODIFY IDS FOR EVENTS WITH SEGMENTS | As desired |
| ENABLE RECORDING OF RECORD EVENTS | As appropriate |
| ENABLE BACK TO BACK RECORD | Enabled |
| RECORDING QUALIFIERS | As appropriate |

**Table 2-1** Louth Device Parameters

| Device Parameter | Setting |
|---|---|
| REPORT ITEMS AUTOMATICALLY DELETED OR RECORDED | As desired |
| DELETE ITEMS AUTOMATICALLY RECORDED FROM DISK AFTER PLAY OUT | Usually disabled |
| AUTOMATICALLY DELETE FROM DISK WHEN FULL | As desired |
| DISK SPACE TO KEEP FREE (MINUTES) | As desired |
| DISK HAS ARCHIVE | Enabled if MediaHub available |
| BACKUP PLAY FROM ARCHIVE SUPPORTED | Disabled |
| CLOSE DISK PORT WHEN NOT IN USE | Disabled |
| DISK SERIAL COMM. PORT NUMBER OF SAME VID DISK | As appropriate |
| CACHE RECORD DISK SERIAL COMM. PORT NUMBER | 0 |

The port number specified in the Louth device parameters corresponds to a port-number-to-port-name mapping in the Louth configuration file (*louth.conf*).

### Signal Configuration

To each controlled video port, you must add a “signal” configuration line which specifies how to map port numbers to port names. For example, to specify that video port 1 corresponds to the video port known as DIVO_4, add the following line:

```
signal 1 DIVO_4
```

Each video port configured on the Louth system needs a similar signal line.

### Configure the Ports on the VCP-Recorder System

To configure the serial ports used by the Louth ADC-100 automation controller, edit the */usr/vtr/config/louth.conf* file on the VCP-Recorder server. For each serial connection, add the following line to this file:

`control` *type* *serial_port* *speed* *[parity]* *video_por frame_rate* *preroll*

where

- *type* is one of the following:
  - rs232 for a RS-232 control connection.
  - rs422 for a RS-422 control connection. The RS-422 connection is available on Origin and Onyx2 servers, only.
- *serial_port* is the serial port number (for example, 1 or 2).

  For example, serial port 2, which may be referred to as tty2 in the hardware inventory (see hinv(1M)) or as /dev/ttyd2 by some software programs, is specified simply by the number 2.
- *speed* is the speed of the connection. The speed must be 38400.
- *parity* is one of the following:
  - 0, indicating no parity
  - 1, indicating odd parity
  - 2, indicating even parity

  The parity must be specified as odd as required by the Louth protocol.

**Note:** Two serial port connections are required if you want to be able to play and record simultaneously. In this case, you must add two lines to the */usr/vtr/config/louth.conf* file, one for each port.

- *video_port* specifies the default video port number to be controlled by this control port.

  A '-' may be used to specify that there is no default port. Otherwise, the numbered port specified here is automatically opened when the system is initialized and must match the port number configured in the DEVICE PARAMETERS for the control port on the Louth system.

- *frame_rate* specifies the frame rate to be used when converting between HH:MM:SS:FF and frame numbers.

  This should match the frame rate of the media being played or recorded by the port.

- *preroll* specifies the number of frames of “disk preroll” used and must match the value configured in the DEVICE PARAMETERS for the control port on the Louth system.

  “Disk preroll” is how far in advance the control system sends commands to the VCP-Recorder. Normally, the control system tells the VCP-Recorder protocol processor what value has been configured, but occasionally this value is delayed. Specifying the value in the Louth configuration file on the VCP-Recorder ensures that the correct value is always used.

## Example Control Connection

The following example defines an RS-232 control connection that uses serial port 2 and specifies odd parity:

```
control rs422 2 38400 1 1 29.97 3
```

Defines control port using RS-422 serial protocol on serial port tty2. The standard bit rate of 38400 and odd parity are used. The default video port is port 1 with a frame rate of 29.97 frames/sec, and there are 3 frames of disk preroll.

*Chapter 3*

# Installing the Vela Research MPEG-2 Decoder

For MPEG2 decoding, VCP-Recorder supports the Vela Research 4-port SCSI-attached decoders. These decoders communicate with the VCP-Recorder server over a Single-Ended, Fast / Wide SCSI bus and decode MPEG2 Transport or Program streams to composite analog video.

The instruction manual that comes with the Vela decoder describes in detail how to connect the Vela decoder hardware and set the SCSI ID.

This chapter describes how to install and configure the Vela decoder in the following sections:

- "Hardware Requirements" on page 27.
- "Connecting the Vela Decoder" on page 28.
- "SCSI Ports" on page 28.
- "Configuring the Vela Decoder" on page 28.

## Hardware Requirements

The Vela decoder requires a single-ended SCSI port. If your server has a mixture of single-ended and differential ports. Be careful to select the single-ended SCSI ports.

The built-in external SCSI ports on the O2 and Origin2000 are single-ended.

The Origin200 does not have a built-in SCSI port. PCI expansion cards are available for both single-ended and differential SCSI. A single-ended SCSI expansion card must be installed.

The M-SCSI XIO card has four SCSI ports: port is auto-sensing for single-ended or differential. Ports 1, 2, and 3 are differential only and cannot be used with the Vela.

VCP-Recorder supports up to three four-port decoders on each SCSI bus. No other SCSI devices should be attached to the bus that is attached to the Vela decoders.

## Connecting the Vela Decoder

The Vela decoders are attached to the SCSI port on the VCP-Recorder system using a standard SCSI-3 cable. Since the decoder does not provide a SCSI loop-through connector, if you are connecting more than one decoder to a single bus, you will need to use a bus-type cable, usually available as a ribbon cable with 4 to 6 SCSI-3 connectors attached.

### SCSI Ports

The Origin2000 base I/O module includes one external single-ended ultra SCSI channel. Additional channels can be added via the M-SCSI four-port Ultra SCSI XIO option (XT-SCSIB-4P), or through the internal PCI adapter XIO option (XT-INT_PCI) together with PCI SCSI adapters.

The Origin200 does not have an external SCSI connector for adding external SCSI devices. Therefore, in order to add external SCSI devices you will need to add an Ultra SCSI PCI adapter to the Origin200. The single-ended Ultra SCSI PCI adapter (PCI-SCSI-Q-SE-1P) should be used.

#### Setting the SCSI ID

Refer to the Vela documentation.

## Configuring the Vela Decoder

Ensure that the *vcp_recorder_eoe_sw.vela* and *.mpeg* subsystems are installed. No configuration is required.

When the system is rebooted after connecting the Vela decoders (make sure they are powered on before rebooting the VCP-Recorder system), the decoders should show up as “Unknown Type 12” devices in the hardware inventory (see hinv(1M)). For example:

```
Unknown type 12: unit 3 on SCSI controller 1
```

The *scsicontrol* command may be used to test the connection by sending a device inquiry to the decoder. To inquire of the device with SCSI id 3 on SCSI bus 1, use the command:

```
% scsicontrol -i sc1d3l0
```

The device should respond as being ready and identify itself as a Vela 4-port decoder.

When VCP-Recorder is started, each of the four ports on each decoder is assigned a port name. The names are sequential, for example, if one decoder is attached, the ports are named vela_0, vela_1, vela_2, and vela_3.

*Chapter 4*

# Installing V-LAN Transmitters

V-LAN is a networked system from VideoMedia for frame-accurately controlling and synchronizing video V-LAN devices, such as videotape recorders. VCP-Recorder includes an interface module which allows it to communicate with a V-LAN transmitter and thereby remotely control a VTR for frame-accurate capture and lay-down of video clips.

Integrating V-LAN and VCP-Recorder is discussed in the following sections:

- "Configuring the Hardware" on page 31.
- "Configuring the VCP-Recorder V-LAN interface" on page 32.

## Configuring the Hardware

Refer to the V-LAN Installation guide for the correct installation of the V-LAN hardware and for connecting the V-LAN transmitter to VCP-Recorder via a serial data connection.

## Configuring the VCP-Recorder V-LAN interface

For each V-LAN device controlled by VCP-Recorder, you enter a line in */usr/vtr/config/vlan.conf*. The general form of the configuration line is:

```
control   type   serial_port   speed   parity   VLAN_node
```

where

- *type* is one of the following:
  - – rs232, for a RS-232 control connection.
  - – rs422, for a RS-422 control connection. The RS-422 connection is available on Origin and Onyx2 servers, only.
- *serial_port* is the serial port number (for example, 1 or 2).

  For example, serial port 2, which may be referred to as tty2 in the hardware inventory (see hinv(1M)) or as */dev/ttyd2* by some software programs, is specified simply by the number 2.
- *speed* is the speed of the connection.

  The speed must match the speed that has been configured on the V-LAN transmitter. The speed is usually 9600 BPS.
- *parity* must be no parity (0).
- *VLAN_node* is a number that identifies the target receiver on the V-LAN network.

  There can be multiple deck-controlling receivers connected to the transmitter. The *VLAN_node* specifies which one VCP-Recorder controls.

The following example defines an RS-422 control connection that uses serial port 2 at a speed of 9600 BPS, no parity, and a V-LAN node of 1:

```
control rs422  2 9600 0 1
```

**PART ONE**

# Managing VCP-Recorder

This part of the book describes how to manage VCP-Recorder. Management tasks include starting and stopping VCP-Recorder, reviewing log messages, installing media clips, and monitoring system stress.

Chapter 5, “Starting and Stopping VCP-Recorder.”

Chapter 6, “Logging.”

Chapter 7, “Managing Clips.”

Chapter 8, “Monitoring the System.”

*Chapter 5*

# Starting and Stopping VCP-Recorder

VCP-Recorder is normally started automatically when the VCP-Recorder server boots. In a development environment, you may choose to disable automatic startup and, instead, start it manually.

While it is possible to invoke VCP-Recorder's main executable, *vvtr*, it is preferable to invoke the VCP-Recorder daemon. The VCP-Recorder daemon, *vtrd*, starts, monitors, and, if necessary, restarts *vvtr*.

This chapter explains how to start VCP-Recorder in the following sections:

- "Starting VCP-Recorder Manually" on page 35.
- "Starting VCP-Recorder at Boot Time" on page 38.
- "Starting and Stopping VCP-Recorder" on page 38.
- "Confirming the Start of VCP-Recorder" on page 38.

## Starting VCP-Recorder Manually

If VCP-Recorder is not started automatically, you can start it manually using the following command:

```
# /usr/vtr/bin/vtrstart
```

*vtrstart* starts *vtrd* which in turn starts *vvtr*, the VCP-Recorder server software.

You should be root when you start *vvtr* to ensure reliable operation. If you are not root, VCP-Recorder starts but writes a warning message to *vtrlog*. For more information about *vtrlog*, see Chapter 6, "Logging."

**Note:** When VCP-Recorder is started manually, in addition to *vtrlog* and *syslog*, its log messages are written to the window from which *vvtr* is invoked.

## Graphics Window

There is an optional graphics port available on O2 that allows you to see the outgoing video in a window, the VCP-Recorder X Window, on the graphics display. This window is sized to display a 625-line video frame with non-square pixels (720x576). If you are playing video with a smaller image size, the image is centered within the window.

**Note:** If you are playing video with non-square pixels (CCIR601 timing), the aspect ratio in the graphics window is incorrect; pixels are square on graphics.

The graphics window is enabled by default on O2. To disable it, edit */usr/vtr/config/device-defaults/mvp* and set *vtr.media.graphics.enabled* to false.

If you wish to use the graphics port, you cannot start VCP-Recorder automatically at boot time. The *vvtr* X Window appears only when you use */usr/vtr/bin/vtrstart* to start the VCP-Recorder.

To disable automatic startup, set

```
% chkconfig vtr off
```

## Start Up Options

The default installation operates normally without changing the start up options. However, the start up options for *vvtr* can be changed by editing */usr/vtr/config/vtrd.conf* and adding them immediately after the program path, */usr/vtr/bin/vvtr*. The line in *vtrd.conf* appears as follows:

```
vvtr * * * * TERM “/usr/vtr/bin/vvtr -l p”
```

TERM is the signal sent by *vtrd*, the daemon, to the *vvtr* processes when they are to be shutdown.

If a process does not shut down using TERM, you might instead send a KILL signal.

**Note:** For the other signals that can be sent, look at */usr/include/sys/signal.h*.

Table 5-1 describes the start up options for VCP-Recorder.

**Table 5-1** Start Up Options

| Option | Description |
|---|---|
| -v *loglevel* | Sets the severity level of the messages that are written to the console log.<br>If this 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 positive or negative, identifies the minimum level of the messages that are written to the log.<br>See on Table 6-2 on page 37 for an explanation of the VCP-Recorder severity levels. |
| -l *logopts* | Sets the log options, which control the format of the messages written to the log. If this option is omitted, each message contains the severity code, a timestamp, the process ID, and the text of the message.<br>The following may be specified for this option:<br>-l l, don’t include the severity code in each log message<br>-l t, don’t include the timestamp in each log message<br>-l p, don’t include the process ID in each log message<br>See “VCP-Recorder Log Messages” on page 45 for more information about the format of the log messages. |
| -p | Specifies that VCP-Recorder should not run at high priority (p). If this option is omitted, VCP-Recorder runs at the highest system priorities.<br>VCP-Recorder must run with high priority to ensure real-time video performance. This option should only be used when it is suspected that a bug is causing a single processor system to hang. (If VCP-Recorder were to go into an infinite loop while running at the highest system priorities, it would lock out all other processes on that processor.)<br>This option should also be used when running a decoder that cannot run in real time. Without this option, the system would continue to run, but all processing, except for VCP-Recorder, would not be able to run while a clip is playing. |

You can also specify any of the log options described in “Logging Options” on page 42.

## Starting VCP-Recorder at Boot Time

You can start VCP-Recorder automatically at boot time by turning on the *vvtr* configuration state as follows:

```
# chkconfig vtr on
```

## Starting and Stopping VCP-Recorder

The commands to start and stop VCP-Recorder from the command line are in */usr/vtr/bin*. If this directory is not defined in your PATH statement, you need to prepend it to your commands.

To start and stop VCP-Recorder, respectively, from the command line, use the following commands:

```
# vtrstart
# vtrstop
```

## Confirming the Start of VCP-Recorder

The easiest way to see whether or not VCP-Recorder is running is by using the *vtrstat* command, as follows:

```
# vtrstat
VCP-Recorder on <host> is running
#
```

***vtrstat*** returns either “`running`” or “`stopped`.”

## What Happens When VCP-Recorder Crashes

Inside */var/adm/crash* directory is a subdirectory for each program that might crash. 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 VCP-Recorder detects a program crash, it renames the core file to core.number, where number is the next, incremental number. This prevents core files from overwriting one another.

You can use debugging tools to examine the core file.

*Chapter 6*

# Logging

Log messages advise you of the status of VCP-Recorder. The preset logging messages are associated with levels of warning, from informational messages, to messages critical to the running of VCP-Recorder. When problems occur, log messages are your first clue for tracking down the problem.

The major tasks involved in managing logging are discussed in the following sections:

- “Configuring Logging” on page 41.
- “Logging Options” on page 42.
- “Managing Log Rollover” on page 45.
- “VCP-Recorder Log Messages” on page 45.

## Configuring Logging

Logging is configured with the start-up options for VCP-Recorder, */usr/vtr/bin/vvtr*. These options are specified in the VCP-Recorder daemon (*vvtrd*) configuration file, */usr/vtr/config/vtrd.conf*. *vvtrd* starts, monitors, and, if necessary, restarts *vvtr*, VCP-Recorder’s main executable.

To modify the VCP-Recorder server logging settings, edit *vtrd.conf* and add or modify the logging options specified in the *vvtr* command string.

For example, to log all log messages up to the Debug2 level in the server log file, */var/adm/vtr/logs/vtrlog*, the *vvtr* entry in *vtrd.conf* should read:

```
vvtr * * * * TERM “/usr/vtr/bin/vvtr -f2
```

For more information about *vtrd.conf*, see “Starting VCP-Recorder at Boot Time” on page 38.

## Logging Options

Whether entered at the command line, or as a line of text in */usr/vtr/config/ vtrd.conf*, the VCP-Recorder daemon command takes the logging options described in Table 6-1.

**Table 6-1** Logging Options

| Flag | Variable | Definition |
|---|---|---|
| -f | log-level | Set the maximum log message priority for the log file to *Info* + *log-level*. The default *log-level* is 0, meaning all log messages up to and including Info priority are written to the log file. |
| -F | log-file | Write the log file to *log-file*. The default is */var/adm/vtr/logs/vtrlog*. For more information, see “Logging Outputs” on page 44. |
| -l | log-options | Omit specified log message fields from log messages. For more information, see “Log level Options” on page 44. |
| -s | log-level | Set the maximum log message priority for SYSLOG to *Info* + *log-level*. The default is 0, meaning all log messages up to and including Info priority are written to SYSLOG. For more information, see “Log Levels” on page 43. |
| -t | | Truncate the log file when it is opened. If this option is not specified, the new log messages are appended to the existing log file. |
| -v | log-level | Set the maximum log message priority for stdout to *Info* + *log-level*. The default is 0, meaning all log messages up to and including *Info* priority are written to stdout. |

The **-v** option is useful only when the VCP-Recorder server, */usr/vtr/bin/vvtr*, is started alone. When the server is started by the VCP-Recorder daemon, as it is when the *vtrstart* command is used, all stdout log messages are discarded.

## Log Levels

**-s** sets the severity level of the messages written to the server log, *vtrlog*. Table 6-2 shows the VCP-Recorder log severity levels and codes, which are listed in decreasing order of severity.

**Table 6-2** Log Levels

| Severity Level | Severity Code | Description |
|---|---|---|
| Emergency | X | Panic condition. |
| Alert | A | A condition that should be corrected immediately, such as a corrupted system file. |
| Critical | C | A critical condition that has system-wide impact, such as a hard device error; immediate action is required. |
| Error | E | A problem that needs correcting, but does not require immediate action. |
| Warning | W | Possible problem, but could be a transient problem that corrects itself. |
| Notice | N | Condition that might require attention, but isn't an error condition. |
| Info | I | Informational message. |
| Debug | *n* | Information message that normally is of use only to engineers for debugging; may be Debug1, Debug2, or Debug3, with Debug3 producing the most 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 the 14th day 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 is an example of a warning message that was written to the log because the user who started the *vvtr* program did not have root permission:

```
W 02-09:00:04.677265 3459 Must be root to ensure reliable operation
```

## Log level Options

**-l** sets the log options, which control the format of the messages written to the log. If this option is omitted, each message contains the severity code, a time stamp, the process ID, and the text of the message.

The following may be specified for this option:

- -l l, do not include the severity code in each log message.
- -l t, do not include the time stamp in each log message.
- -l p, do not include the process ID in each log message.

Level, process ID, and time stamp are always omitted from the log messages in the system log, *SYSLOG*.

## Logging Outputs

**-F** specifies where logging is output. Output options include:

- stdout
- server log file (*/var/adm/vtr/logs/vtrlog*)
- system log (*/var/adm/SYSLOG*)

By default, critical log messages go to the system log, *SYSLOG*. These messages are also logged to the server log file (*/var/adm/vtr/logs/vtrlog*) as are all messages up to and including the Info level.

Informational log messages go to the server log file, *vtrlog*.

Log messages go to *stdout* by default only when you start VCP-Recorder manually. In that case, the log messages go to the window in which VCP-Recorder was invoked.

If VCP-Recorder is invoked automatically at boot time, you should configure the logging level for the server log file (**-f** option) as desired to help troubleshoot VCP-Recorder problems.

## 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 problem of having a list of log messages too long to easily handle.

SYSLOG is rolled over automatically by a cron job. Often the cron job:

1. Renames the current log file.
2. Sends a SIGHUP. (killall -HUP syslogd) command to the logging daemon, which causes it to start logging into a new log file.

### Managing vtrlog

*vtrlog* is the only server log file you need to manage. In general, to manage it, you build a cron script that mimics the actions of the SYSLOG cron script. The details of the cron script are left to the discretion of the system administrator; VCP-Recorder has no requirements for rollover parameters.

When the VCP-Recorder daemon (*vtrd*) or server (*vvtr*), receives a *SIGHUP* signal, it reopens the server log file. If you rename or remove the server log file and then send a *SIGHUP* signal, logging starts in a new, empty log file.

## VCP-Recorder Log Messages

Each VCP-Recorder log message has the following format:

*c dd-hh:mm:ss.mmmmmm pppp <log message>*

where

- *c* is the severity code of the message.
- *dd* is the day of the month.
- *hh:mm:ss.mmmmmm* is a time stamp that indicates when the message was written.
- *pppp* is the process ID of the process that wrote the message to the log.
- *log message* is the actual text message.

*Chapter 7*

# Managing Clips

Clips are segments of audio and video media. Managing clips involves adding and removing them from VCP-Recorder.

Managing clips is discussed in the following sections:

- "VCP-Recorder Media Formats and Types" on page 47.
- "Adding Clips" on page 49.
- "Removing Clips" on page 52.
- "Exporting VCP-Recorder Clips" on page 52.

**Note:** For information about configuring the clip cache, see "Configuring the Clip Cache" on page 14.

## VCP-Recorder Media Formats and Types

VCP-Recorder uses two different formats and two types of media.

### Media Format

VCP-Recorder uses two different formats for storing digital media on its clip cache file system(s):

- vframe format - for intraframe media such as JPEG.
- stream format - for interframe media such as MPEG2.

### vframe Format

The vframe (short for variable-size frame) format is specific to VCP-Recorder. The video and audio data for each frame is stored contiguously in the media file. A separate index file contains header information which describes the encoding parameters of the video and audio data, and a frame map which stores the offsets of the video and audio data for each frame within the media file.

The vframe format is used for recording and playback of intraframe media, including uncompressed video as well as Rice-coded and JPEG-compressed video.

The *vtrvfutil* command-line utility provides various analysis and update functions for the vframe format. See the vtrvfutil man page for further information.

### Stream Format

The stream format consists of a media file, which is simply the native media bitstream, and an index file, which maps random access cueing points to the appropriate offset in the bitstream.

The stream format is used for playback of MPEG2 media.

## Media Types

VCP-Recorder also accepts two types of media:

- Intraframe
- Interframe

### Intraframe

In intraframe media, the video data for each frame is self-contained and does not depend on the data from neighboring frames. Examples of intraframe media include JPEG, Rice, and uncompressed video.

### Interframe

In interframe media, the compression techniques used for some video frames may require data from neighboring frames in order to decompress the video frame.

# Adding Clips

Adding clips is a two step process:

1. Transfer the clip to the file system.
2. Tell VCP-Recorder it is there.

## Transferring Clips to File Systems

There are two types of files systems:

- Non-real time.
- Real time.

To guarantee real time I/O rates, you must use a real time file system.

### Non-Real Time File Systems

Transferring files to a non-real time file system is as easy as using *ftp*, *cp*, or *rcp*.

### Real Time File Systems

Only real time file systems guarantee real time I/O rates.

A real time file system is often defined as a subvolume in an XFS file system where media is stored. The location of the subvolume is often *xls/xlv/*. Data is read normally from a real-time file system, but the standard utilities, such as *ftp* and *cp*, can not be used to write the media data.

## Clip Alignment

In single-disk systems, media files are aligned only with the file system's block size so that only one I/O operation is needed to access the data.

In multi-disk, RAID, striped systems, the media files must also be aligned with the stripes.

When VCP-Recorder is used to record media, it automatically places the media data in the correct, aligned locations on the disk.

When media data is transferred to VCP-Recorder it often must be aligned with the disk's blocks and stripes. Whether or not media data must be aligned on your disks depends on the media type of the media file:

- Intraframe media data requires alignment.
- Interframe media data does not require alignment.

The media data (video and audio for each frame) for an intraframe media clip must be aligned on natural file system and disk drive boundaries to enable efficient read and write access to the clip.

### Degree of Alignment

Frame-oriented media data in an intraframe clip is aligned along 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. VCP-Recorder will not do a read or write operation to the clip media file which crosses a major alignment boundary.

### Minor Alignment

The minor alignment matches the greater of the file system block size, or the system memory page size. On O2, the minor alignment is usually 4KB, unless a larger file system block size was used to construct an XFS file system. On Origin, the minor alignment is usually 16KB, unless a larger file system block size was used.

### Major Alignment

The major alignment matches the stripe size of the disk volume which holds the clip cache file system.

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. Either make a real-time file system on the RAID and set the real-time extent size of the XFS file system to the desired I/O operation size, or add a configuration line to */usr/vtr/config/vtrfsinfo.conf*.

**Note:** *vtrfsinfo.conf* is not created by VCP-Recorder so you first must create the file.

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

### Realigning a Clip

A vframe clip is automatically aligned if you use *vtrvfutil* to copy the clip, as follows:

```
% vtrvfutil OriginalClipName NewClipName
```

*vtrufutil* automatically stores the media files in */usr/vtr/clips/*.

To copy a vframe clip from outside the clip cache, use the following syntax:

```
% vtrvfutil -i file=<media file>,index=<index file> - NewClipName
```

*vtrufutil* can also be used to realign a clip in-place, as follows:

```
% vtrvfutil -c realign ClipName
```

See the *vtrufutil* man page for a complete list of the options.

## Notifying VCP-Recorder

After transferring the media files, you must notify VCP-Recorder of the file by adding the clip, as follows:

```
# vtrclip add clipName
```

*clipName* is the name of the clip file.

## Removing Clips

There are two ways of removing clips from VCP-Recorder.

- VCP-Recorder Clip Manager (mcclips)
- Command line

For more information about using the VCP-Recorder Control panel to remove a clip, see the *VCP-Recorder User's Guide*.

### Command Line Removal

You can completely remove a clip from VCP-Recorder using the following command:

```
# vtrclip rm clipName
```

*clipName* is the name of the clip file.

## Exporting VCP-Recorder Clips

You can use the *vtrvfutil* command to add Quicktime formatting information to a clip using the following command:

```
# vtrvfutil -c makeqt clipName
```

*clipName* is the name of the clip file.

*vtrvfutil* makes the clip readable by SGI digital media tools, such as *dminfo*, *dmconvert*, and mediaplayer, or third-party tools that use the SGI Movie Library.

*Chapter 8*

# Monitoring the System

Once VCP-Recorder is installed and running you can monitor its operation using the tools mentioned in this chapter. Two tools, *vtrstat* and *mcstat*, work directly with VCP-Recorder. You can also use standard Silicon Graphics IRIX tools to monitor various system resources.

There are two tools that monitor VCP-Recorder directly:

- vtrstat
- mcstat

*mcstat* is the graphical version of *vtrstat*. Both tools monitor VCP-Recorder operations.

This chapter describes how to monitor VCP-Recorder in the following sections:

- “vtrstat” on page 53.
- “mcstat” on page 54.
- “IRIX Tools” on page 55.

## vtrstat

*vtrstat* is a command line tool that tells you:

- Whether or not VCP-Recorder is running.
- Which units are open, using the **-units** option.
- Which media ports are available, using the **-ports** option.

Example 8-1 shows an example output of *vtrstat*.

**Example 8-1** vtrstat Output

```
kruger 87# /usr/vtr/bin/vtrstat -units
VCP-Recorder on kruger is running.
```

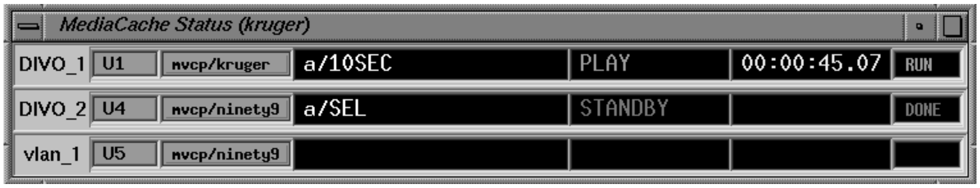
```
Unit  Owner        Port    Clip       Function Location
----------------------------------------------------------
U1    louth        DIVO_0  d/REC12    STOP     01:04:30.05
U2    louth        DIVO_0  *          STOP     *
U3    louth        DIVO_1  d/rPOP     CUE      00:00:00.00
U4    louth        DIVO_1  d/rSEL     STOP     00:00:00.00
U5    louth        DIVO_1  d/REC01    STOP     01:04:30.06
U6    louth        DIVO_1  *          STOP     *
U7    louth        DIVO_0  d/rSEL     CUE      00:00:00.00
U8    louth        DIVO_0  d/rSTOK    STOP     00:03:30.01
U9    louth        DIVO_7  d/rJOR     CUE      00:00:00.00
U10   louth        DIVO_7  d/rSEL     STOP     00:00:00.00
U11   louth        DIVO_3  d/rPOP     CUE      00:00:00.00
U12   louth        DIVO_3  d/rSTOK    PLAY     00:03:02.29
U13   louth        DIVO_4  d/rSEL     CUE      00:00:00.00
U14   louth        DIVO_4  d/rSTOK    STOP     00:00:00.00
U15   louth        DIVO_5  d/rSEL     CUE      00:00:00.00
U16   louth        DIVO_5  d/rSTOK    STOP     00:03:30.01

kruger 88# /usr/vtr/bin/vtrstat -ports
VCP-Recorder on kruger is running.

#  Port    Type    Description
----------------------------------------------------------
0  vlan_1  Deck    VLAN Deck Control
1  DIVO_0  Video   SGI XT-DIVO Digital Video Option
2  DIVO_1  Video   SGI XT-DIVO Digital Video Option
3  DIVO_3  Video   SGI XT-DIVO Digital Video Option
4  DIVO_4  Video   SGI XT-DIVO Digital Video Option
5  DIVO_5  Video   SGI XT-DIVO Digital Video Option
6  DIVO_6  Video   SGI XT-DIVO Digital Video Option
7  DIVO_7  Video   SGI XT-DIVO Digital Video Option
```

## mcstat

*mcstat* is the graphical equivalent of *vtrstat*. *mcstat* tells you:

- Whether or not VCP-Recorder is running.
- Which units are open.
- Which media ports are available.

Figure 8-1 shows an example of *mcstat*'s output.



MediaCache Status (kruger)

| | | | | | | |
|---|---|---|---|---|---|---|
| DIVO_1 | U1 | mvcp/kruger | a/10SEC | PLAY | 00:00:45.07 | RUN |
| DIVO_2 | U4 | mvcp/ninety9 | a/SEL | STANDBY | | DONE |
| vlan_1 | U5 | mvcp/ninety9 | | | | |

**Figure 8-1** mcstat

For more information about mcstat, see the *mcstat User's Guide*.

# IRIX Tools

Silicon Graphics provides a compendium of monitoring tools that can be used with VCP-Recorder, including

- sar—system activity reporter, reports operating system activity.
- gr_osview—graphical system monitor, graphically displays real-time usage of certain system resources.
- pcp—Performance Co-Pilot, serves as an interface for existing reporting tools, such as graphing performance data over time.

Before exporting media data recorded by VCP-Recorder to IRIX tools, you must manipulate the data. For more information about exporting media data, see Chapter 7, "Managing Clips," "Exporting VCP-Recorder Clips" on page 52.

## Using sar

*sar*, the System Activity Reporter, is an activity counter. The command line options allow you to specify the kinds of activities you want measured. You can, for example, display disk utilization.

To use *sar*, use the following procedure:

1. Enable *sar* using the following command:

   ```
   # chkconfig sar on
   ```

2. Reboot your system.

   When your system reboots, the *sar* data collector starts.

3. Specify the system activity you want *sar* to measure, for example, to measure disk utilization, use a command similar to the following:

   ```
   # sar -d 2 10
   ```

   The **-d** command line option specifies the display of disk utilization. The numbers, 2 and 10, specify how often you want to take reports. In this example, 10 reports are taken every 2 seconds.

**Note:** For a complete list of *sar* command line options, see the *sar* man page.

Example 8-2 shows an example output of *sar*.

**Example 8-2** SAR's Output

```
kruger 100# sar -d 2 10

IRIX64 kruger 6.4 02121744 IP27    12/19/1997

11:15:10     device %busy  avque  r+w/s  blks/s    w/s wblks/s
avwait  avserv
11:15:12
dks2d1     8    1.0      3   12002      0       0     0.0    28.3
dks3d69    8    1.0      1   12257      0       0     0.0    53.3
dks0d1     8    5.4      7     120      7     120    49.3    11.3
dks1d1     0    0.0      0       0      0       0     0.0     0.0
dks0d2     0    0.0      0       0      0       0     0.0     0.0
dks1d2     0    0.0      0       0      0       0     0.0     0.0
dks0d3     0    0.0      0       0      0       0     0.0     0.0
dks1d3     0    0.0      0       0      0       0     0.0     0.0
dks0d4     0    0.0      0       0      0       0     0.0     0.0
dks1d4     0    0.0      0       0      0       0     0.0     0.0
dks0d5     0    0.0      0       0      0       0     0.0     0.0
dks1d5     0    0.0      0       0      0       0     0.0     0.0
dks0d6     0    0.0      0       0      0       0     0.0     0.0
dks1d6     0    0.0      0      16      0      16     0.0     0.0
```

## Using gr_osview

*gr_osview* provides a graphical display of system resources usage. This display provides a real-time window into the overall operation of the system, as shown in Figure 8-2.

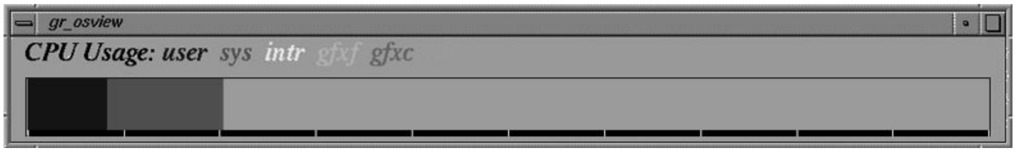


**Figure 8-2** gr_osview

For more information about *gr_osview*, see its man page.

## Using Performance CoPilot (PCP)

Performance Co-Pilot (PCP) provides a suite of tools that co-operate to deliver distributed, integrated performance-monitoring and performance-management services across a spectrum of performance domains, as shown in Figure 8-3.

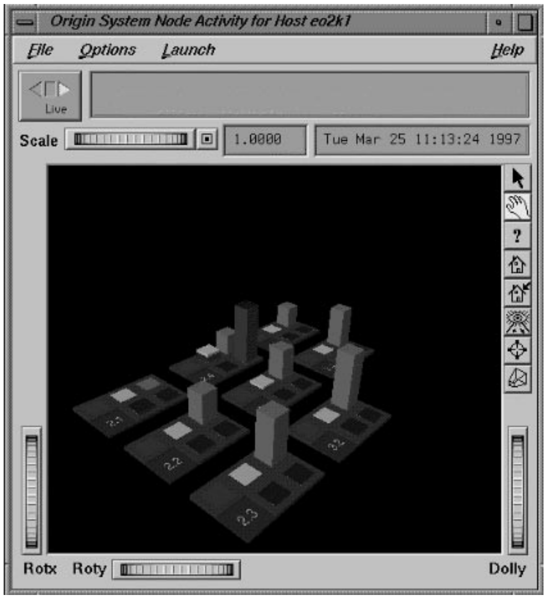


**Figure 8-3** Performance CoPilot

For more information about Performance CoPilot, see its man page.

# Index

## Tell Us About This Manual

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

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

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

Please send the title and part number of the document with your comments. The part number for this document is 007-3622-001.

Thank you!

## Three Ways to Reach Us

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

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