Video Server Toolkit
Installation and System Administrator's Guide

Contents

List of Examples

List of Figures

List of Tables

About This Guide

Video Server Toolkit (VST) is Silicon Graphics' broadcast-quality video playback, edit, and record engine that unifies Silicon Graphics Origin servers and digital media components. VST allows application developers to create powerful, high-performance solutions for broadcast playout and editing. This tool enables (supported) automation systems and developer applications to trigger video for playout though (supported) video output devices on a Silicon Graphics workstation or server.

Note: VST is the new name for what used to be called VCP-Recorder.

For more information about using VST, see the *Video Server Toolkit Developer's Guide.*

The *Video Server Toolkit Installation and System Administrator's Guide* describes how to:

- Set up the system hardware and complete pre-VST-installation software installation and configuration (Part 1)

- Install and configure the VST software (Part 2)

- Monitor the VST system (Part 3)

This chapter describes this book in the following sections:

- "What This Document Contains" on page xxiv

- "Who Should Read This Document" on page xxvi

- "Related Documentation" on page xxvii

- "Conventions Used in This Document" on page xxviii

What This Document Contains

This document contains three parts:

- Part I specifies tasks to be completed before installing the VST.

- Part II describes how to install and configure VST and all of the hardware and software you might use in conjunction with VST.

- Part III describes how to manage your VST system.

Part I, "System Setup"

- Chapter 1, "Setting Up the Hardware and the System," explains the steps you need to take to set up your system so that VST can run on it.

- Chapter 2, "Setting Up the Clip Cache for Video Server Toolkit," describes how to create a real-time filesystem to use with the VST clip cache.

Part II, "Installing and Configuring Video Server Toolkit"

- Chapter 3, "Installing Video Server Toolkit," provides a step-by-step description of installing the VST software.

- Chapter 4, "Configuring Video Server Toolkit," describes the configuration files and settings for VST.

- Chapter 5, "Time Code Readers," provides a very brief discussion of time code and how to configure VST to work with the time code generators.

- Chapter 6, "Installing IRIS FailSafe," describes how to install FailSafe, which is a monitoring system that switches system functions from a primary to a secondary server when the primary server fails.

- Chapter 7, "Configuring and Using the Sony or P2 Protocol," describes how to configure VST to use the Sony protocol.

- Chapter 8, "Diaquest," describes how the VST can be used with Diaquest software to control an external media device.

- Chapter 9, "Installing the Louth Automation Controller," describes how to configure the Louth automation controller to work with VST.

- Chapter 10, "Configuring and Using V-LAN Transmitters," describes how to integrate V-LAN and VST.

- Chapter 11, "Installing the Vela Research MPEG-2 Decoder," describes how to install and configure the Vela decoder.

- Chapter 12, "Installing the DVB-ASI Adapter Board," describes how to install and configure the DVB-ASI encoder/decoder board.

Part III, "Managing Video Server Toolkit"

- Chapter 13, "Starting and Stopping VST," explains how to start VST.

- Chapter 14, "Logging," describes the diagnostic log messages output by VST.

- Chapter 15, "Monitoring the System," describes how to monitor VST.

- Chapter 16, "Troubleshooting," lists an assortment of common problems and their solutions.

- Appendix A, "Example Filesystem and Plexing Setup," describes how to set up a highly available file system.

- Appendix B, "Video Server Toolkit Controls," describes the variables, called controls, that help configure the system.

- Appendix C, "Automating Video Server Toolkit Commands Using Perl," discusses the */usr/vtr/lib/perl/MVCP.pm* Perl module, which, among other things, automates back-to-back recording.

- Appendix D, "RS-422 Pinouts," describes the pinouts for the RS-422 Mini-DIN8 to DB9 adapters for Silicon Graphics and Apple Macintosh cables.

A glossary and index follow the appendices.

HTML versions of the VST books are installed at URL:

- http://*hostname.domain*/VCPRecorder/DevelopersGuide

- http://*hostname.domain*/VCPRecorder/InstallationGuide

Who Should Read This Document

This document is written for VST system integrators and administrators responsible for installing and monitoring the running of VST.

Man Pages

VST contains the following man pages:

- vcp-recorder (Video Computing Platform Recorder)
- vtrstart(1)—for startup
- vtrstop(1)—for shutdown
- vtrstat(1)—for status
- vtrclip(1)—for clip management
- mcpanel(1)—for media control panel
- mcclips(1)—for clip manager
- mcstat(1)—for status display
- mccompstats(1)—for compression monitor
- vcp-recorder-controls(5)—for VCP-R controls
- mvcp(5)—Multiport Video Computer Protocol
- vvtr(1)—for VCP-R server
- vtrd(1)—for VCP-R daemon
- vtrvfutil(1)—for VCP-R vframe clip utility

You can list the VST man pages after installing them by entering the following command:

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

Related Documentation

Refer to the following documents for related information:

- *Video Server Toolkit Developer's Guide* (part number 007-3620-*nnn*) for information about using VST to play and record digital media data; and store the data in, and retrieve it from, an *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.

- *StudioCentral Developer's Guide*, for information regarding the StudioCentral archiving system.

- *DIVO Owner's Guide*, for information about DIVO.

- *Origin200 Owner's Guide* and *Origin200 Maintenance Guide* for information about the Origin200.

- *Origin2000 Owner's Guide* and *Origin2000 Maintenance Guide* for information about the Origin2000.

- *Digital Media Programmer's Guide* for information about Digital Media and RAD.

- *FibreChannel Owner's Guide* for information about the GIGAchannel and RAIDs.

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."

System Setup

This part of the book explains how to set up your system before you install and configure Video Server Toolkit (VST).

Chapter 1, "Setting Up the Hardware and the System," explains the steps you need to take to set up your system so that VST can run on it.

Chapter 2, "Setting Up the Clip Cache for Video Server Toolkit," describes how to create a real-time filesystem to use as the VST clip cache.

Setting Up the Hardware and the System

This chapter explains, in the following sections, the steps you need to take to set up your system so that Video Server Toolkit (VST) can run on it:

- "Supported Hardware" on page 3

- "Hardware Installation" on page 7

- "Downgrading from IRIX 6.5 to IRIX 6.4 and Installing the Miniroot" on page 9

- "Initial System Administration" on page 16

After completing these steps, you are ready to create a real-time filesystem, discussed in the next chapter, Chapter 2, "Setting Up the Clip Cache for Video Server Toolkit."

Note: You do not have to set up a real-time filesystem for VST to work.

Supported Hardware

VST works with the following Silicon Graphics servers and workstation:

- O2 workstation

- Origin200 server with GIGAchannel and DIVO, or with a DVB-ASI board, or with a VELA board

- Origin2000 server

Port Configuration

Table 1-1 lists the port configurations for these servers.

Table 1-1   Port Configurations

Machine	Port Configuration
Origin200 GIGAchannel (single module)	6-1/2 PCI slots and 5 XIO slots; one XIO slot can be used for 100BaseT XIO, which has 6 serial ports.
Origin2000 (one node card; 2 processors)	Either 5 available XIO slots (no PCI slots), or 4 XIO slots with 3 PCI slots.
Origin2000 (two node cards; 4 processors)	Either 11 XIO (no PCI) or 10 XIO with 3 PCI.

External Devices

VST can interact with a variety of external devices, as shown in Figure 1-1.
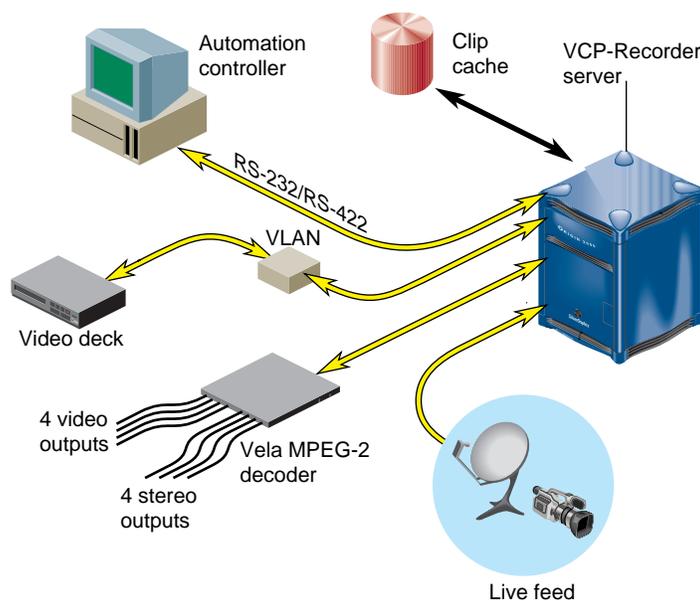
Figure 1-1   VST with External Devices

Part of the installation of VST includes configuring it to work with these devices.

VST works with the following external devices:

- Devices that control VST

    - Sony protocol controller; see Chapter 7, "Configuring and Using the Sony or P2 Protocol."

    - Louth Automation Controller; see Chapter 9, "Installing the Louth Automation Controller."

    - Odetics devices;

- Devices VST controls
    - Videotape recorders; V-LAN Transmitter; see Chapter 10, "Configuring and Using V-LAN Transmitters."
    - Diaquest software, see Chapter 8, "Diaquest."
- Devices through which media is played or recorded
    - Vela Research MPEG-2 Decoder; see Chapter 11, "Installing the Vela Research MPEG-2 Decoder."
    - DVB-ASI; see Chapter 12, "Installing the DVB-ASI Adapter Board."

Disk Storage Options

On Origin servers, there are two storage options:

- UltraSCSI; using four drives gives a bandwidth of 32 MB/sec.
- Fibre channel drives with a bandwidth of 44 MB/sec.

Bandwidth Considerations

A stream is a video/audio program. The number of steams a server can record or play at once depends on the storage option and the compression type.

For example, if you assume 8-bit, 4:2:2 Rice variable compression is, at worst, 10.3 MB/sec, an UltraSCSI system can handle two Rice-compressed video streams (32 MB/10.3 MB = 2 streams) or eight DVCPRO-compressed video streams (32 MB/3.4 MB = 8 streams).

A fibre channel system can handle three (44 MB/10.3 MB = 3 streams) Rice-compressed video streams or eleven (44 MB/3.4 MB = 11 streams) DVCPRO-compressed video streams.

Compression Formats

The bandwidth determines the number of streams of video that can be played or recorded at once. Each compression format requires a different bandwidth, as follows:

- 8-bit 601 video uncompressed—20.6 MB/sec

- Rice-encoded—10.3 MB/sec

- DVCPRO—3.4 MB/sec

- JPEG—typically from 1 MB/sec to 8 MB/sec

- MPEG-2 MP@ML—typically from 3Mbits/sec to 8Mbits/sec

Because the Rice-coded format is lossless, there are few reasons to use uncompressed video unless the video files or frames will be used by other programs, such as an editing application.

Running uncompressed video on an O2 is not supported in version 1.1 of VST.

Hardware Installation

1. Install hardware using instructions from the manual(s):

   - *Origin200 and Origin200 GIGAchannel, Owner's Guide*

   - *Origin200 and Origin200 GIGAchannel, Maintenance Guide*

   - *XIO Digital Video Option (DIVO) Board, Owner's Guide*

   - *PCI Digital Audio Option Board, Installation Guide*

   - *Fibre Channel XIO and PCI Option Board Owner's Guide*

   - *Origin FibreVault and Fibre Channel RAID Owner's Guide*

   - *IRIS 4-Port Fast Ethernet Adapter with Asynchronous Serial XIO Board, Installation Instructions*

2. Connect the system to the network using a 10-Base-T ethernet cable between the RJ-45 connector on the Origin200 Master module and a network port.

   Note: A name and IP address must be assigned to the system to connect it to a network

3. Connect system console (RS-232/RS-422 Serial) port on the Origin200 Master module to a serial port of another system or an ASCII terminal using a serial cable.

    See instructions for *Using a Silicon Graphics Workstation as the System Console* or *Attaching the System Console in the Origin200 Owner's Guide.*

4. Log in to the system console using the serial port on another system. Make sure the serial port on the other system is not being used for another purpose, such as its own system console.

    Use the following command if the other system is a Silicon Graphics machine.

    ```
    cu -s 9600 -l ttyd2
    ```

Installing Additional Hardware

Install any hardware and associated driver software related to VST, such as VLAN, Vela, DVB-ASI, or DIVO boards. When you install VST, 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 system, but required for using VST.

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).

An Origin2000 needs a DIVO (Digital Video Option), Vela, or DVB-ASI board.

VST supports an Origin200 with a DIVO card with a GIGAchannel expansion box, or a Vela board, or a DVB-ASI board.

Downgrading from IRIX 6.5 to IRIX 6.4 and Installing the Miniroot

A miniroot installation is an installation method that relies on services in the system's programmable read-only memory (PROM) to transfer special installation tools from the CD (or distribution directory) to the system disk. It requires that you shut down the system and boot from a CD or distribution directory.

VST works only with IRIX 6.3 or 6.4. If you are using IRIX 6.5, you must downgrade it to 6.4.1 using the following instructions, or the instructions included with the IRIX 6.4.1 CDs.

Overview of Installation Procedure

In the following installation, you complete the following steps:

1. Load the miniroot from the IRIX 6.4 patch CD.
2. mkfs /dev/dsk/dks0d1s0.
3. Install IRIX 6.4 from the OS CD.
4. Install IRIX 6.4 Applications from the Apps CD.
5. Install IRIX 6.4 patches.
6. Restart the system.
7. Install further applications, patches, and non-Silicon Graphics software from multiuser mode.

Pre-Installation Procedure

Use the following procedure to set up your filesystem.

1. If the system is not brand new and you have important data on the disk, perform a full system backup, using the method you normally use for backups.

2. Restart the system using the following command:

   ```
   # init 0
   ```

   The system displays the System Maintenance Menu from the local host.

   ```
   System Maintenance Menu

   1) Start System
   2) Install System Software
   3) Run Diagnostics
   4) Recover System
   5) Enter Command Monitor
   ```

3. Enter **2**, the Install System Software option.

   The system responds:

   ```
                     Installing System Software...

   Press <Esc> to return to the menu.

   1) Remote Tape  2) Remote Directory  X) Local CD-ROM  X) Local Tape

   Enter 1-4 to select source type, <esc> to quit,
   or <enter> to start:
   ```

4. If you have a local CDROM, enter **x**.

   The system responds:

   ```
   Enter 1-4 to select source type, a to select the source, <esc> to
   quit,
   or <enter> to start:

   Insert the installation CD-ROM, then press <enter>:
   ```

   4 a. Press the Enter key and proceed to step 6.

5. Alternately, if you are using a remote CDROM:

5 a. On the remote host edit the */etc/inetd.conf* file to allow the remote system to boot from the CDROM directory by appending */CDROM* to the configuration line, as shown:

```
tftp    dgram   udp     wait    guest   /usr/etc/tftpd  tftpd -s
    /usr/local/boot /CDROM
```

5 b. Restart inetd on the remote system:

```
/etc/killall -HUP inetd
```

5 c. Enter **2**, the Remote Directory option.

The system responds:

```
Enter the name of the remote host:
```

5 d. Enter the name of the remote host (foo used in this example).

The system responds:

```
Enter the remote directory: /CDROM/dist
```

5 e. Enter the name of the remote directory. (*/CDROM/dist* is used in this example).

```
1) Remote Tape  2)[Remote Directory]  X) Local CD-ROM  X) Local Tape

    *a) Remote directory /CDROM/dist from server foo.

Enter 1-4 to select source type, a to select the source, <esc> to
quit,

or <enter> to start:
```

6. Press the Enter key to load the miniroot.

The system responds:

```
Copying installation program to disk.
10% ......... 20% ......... 30% ......... 40% ......... 50%
60% ......... 70% ......... 80% ......... 90% ......... 100%
Copy complete

IRIX Release 6.4 IP27 Version 02121744 System V - 64 Bit
Copyright 1987-1997 Silicon Graphics, Inc.
All Rights Reserved.

root on
/hw/module/1/slot/MotherBoard/node/xtalk/8/pci/0/scsi_ctlr/0/target/
1/lun/0/disk/partition/1/block ; boot swap file on /dev/swap swplo
55000

Creating miniroot devices, please wait...

Current system date is Mon Feb 01 11:02:16 PST 1999

Mounting file systems:

    /dev/miniroot           on  /
    /dev/dsk/dks0d1s0       on  /root
Invoking software installation.
```

7. Clean the root filesystem using mkfs by entering `13` to access the Administration menu.

The system displays:

```
Administrative Commands Menu

 1. set [preferences]            List all preferences or
                                    set/clear a preference
 2. date [args]                  Display or set the system date
 3. files [names]                List files in subsystems
 4. space                        List disk usage information
 5. recalculate                  Recalculate space required for
                                    installation
 6. sh [cmd]                     Escape to a shell or run a
                                    command
 7. shroot [cmd]                 Escape to a chrooted shell or
                                    run a command
 8. relocate [prod [disk] ]      Relocate product to a different
                                     disk
 9. mount [fsname] [dir]         Show mounted filesystems or
                                    mount new ones
10. umount [fsname]              Unmount a filesystem
11. mkfs [blockdevice]           Make a new filesystem
12. sethostname                  Set name of current host, for
                                    networking
13. sethostipaddr                Set host ipaddr - Internet
                                    address
14. save filename                Save selection to file
15. load filename                Load selection from file
16. rqs                          Perform rqs(1) processing
17. config [changed]             List all or modified config
                                    files
18. hardware                     List machine-specific hardware
                                    information
19. updatekeepfile               Add N(ew), unselected
                                    subsystems to keepfile
20. return                       Go to the Inst Main Menu

Admin>
```

8. Enter `11` on the Administration menu of the miniroot Inst program. The system displays:

```
** Clean Disks Procedure **

        If you agree to it, this procedure will clean your disks,
        removing all data from the root and (if present) the user
        file systems.

        Boot device partitions zero (0) and, if present, six (6)
        will be erased (using mkfs).  This will destroy all data on
        hem.  These partitions will then be remounted under /root
        and (if present) /root/usr.

        If you have data on these file systems you want to save,
        answer "no" below, and backup the data before cleaning
        your disks.

        Any other file systems or logical volumes will be unmounted
        and forgotten about until you choose to reconfigure and
        remount them.


      Are you sure you want to clean your disks ?
                { (y)es, (n)o, (sh)ell, (h)elp }: y

WARNING: There appears to be a valid file system on
/dev/dsk/dks0d1s0 already.
Making a new file system will destroy all existing data on it.

Make new file system on /dev/dsk/dks0d1s0?
{ (y)es, (n)o, (sh)ell, (h)elp }:
```

9. Enter **y.**

    The system displays:

    ```
    Doing: mkfs -b size=4096 /dev/dsk/dks0d1s0
    meta-data=/dev/dsk/dks0d1s0      isize=256     agcount=9,
    agsize=243268 blks
    data     =                       bsize=4096   blocks=2189410,
    imaxpct=25
    log      =internal log           bsize=4096   blocks=1000
    realtime =none                   extsz=65536  blocks=0, rtextents=0

    Mounting file systems:

        /dev/miniroot            on  /
        /dev/dsk/dks0d1s0        on  /root
    Admin>
    ```

10. Enter **quit**.

    The Inst Main Menu displays.

11. Enter **from**.

12. When the Inst prompt appears, eject the recommended patches for IRIX 6.4
    CD-ROM, load the IRIX 6.4 for Origin operating system CD-ROM, and enter **from
    /CDROM/dist**:

    ```
    Inst> from /CDROM/dist
    ```

13. When prompted for another distribution, enter **done**:

    ```
    Install software from: [/CDROM/dist] done
    ```

    The system returns an Inst prompt.

14. To create a real-time filesystem, you need the following software:

    ```
    eoe.sw.xfs          01/28/99  XFS Filesystem
    eoe.sw.xfsrt        01/29/99  XFS Realtime & Guaranteed-Rate
                                       Support
    eoe.sw.xlv          01/28/99  XLV Volume Manager
    eoe.sw.xlvplex      01/28/99  XLV Volume Manager Plex Support
    ```

    Enter the following commands at the Inst prompts:

    ```
    Inst> install default
    Inst> i eoe.sw.xfs eoe.sw.xfsrt eoe.sw.xlv eoe.sw.xlvplex
    Inst> go
    ```

15

15. Load the IRIX 6.4. Applications CD and enter **`from /CDROM/dist`**, then enter **`install default`**, then **`go`**:

    ```
    Inst> from /CDROM/dist
    Inst> install default
    Inst> go
    ```

16. Reload the IRIX 6.4 patches CD and install the patches:

    ```
    Inst> from /CDROM/dist
    Inst> install installable
    Inst> keep same
    Inst> go
    ```

    Warning: **You must install the IRIX 6.4 patches during the miniroot install before restarting the system; not doing so may cause the system to hang during the first IRIX 6.4 boot.**

17. When Inst indicates the installation is finished, enter **`quit`**.

    Post installation processes begin. When they are complete you see:

    ```
    Ready to restart the system. Restart?
    Enter y to restart.
    ```

18. Restore any user files, modified system files, or other files you need from the backup created in step 1.

19. Install any non-IRIX patches and required software. Make sure you have the IRIX 6.4 versions of the patches or software. You may need to install further patches from the patch CD.

Initial System Administration

1. Name the system.

   `vi /etc/sys_id`

2. Add system name and IP address to the hostname-address database.

   `vi /etc/hosts`

3. Set IP address in non-volatile RAM; also, set system up for automatic restart.

   `nvram netaddr <IP Address>`
   `nvram AutoLoad Y`

4. Set network address mask, as appropriate for the network.

    `vi /etc/config/ifconfig-1.options`

5. Set default system time zone.

    `vi /etc/TIMEZONE`

6. Set password for user root, if necessary

    `passwd`

7. Reboot the system.

    `reboot`

Setting Up the Clip Cache for Video Server Toolkit

Video Server Toolkit (VST) holds its media assets in the clip cache. The clip cache is maintained on one or more filesystems across one or more disks or RAIDs. This chapter describes how to use the available disks and RAIDs as the VST clip cache by creating the appropriate XFS filesystem(s) on them.

It is beyond the scope of this book to explain in detail the intricacies of the tools, including XFS, XLV, and GRIO, you use to set up your filesystem. For information about any of these tools, see the *IRIX Admin: Disks and Filesystems* book.

For a complete filesystem setup example, see Appendix A, "Example Filesystem and Plexing Setup."

For optimal performance, use the following recommendations:

- All RAIDs used should be RAID Level 3.
- Real-time filesystems should be used whenever possible so that VST can use Guaranteed Rate I/O (GRIO). GRIO provides highly reliable video streaming.
- If multiple disks or RAIDs are available, use XLV to stripe data across the disks.

  Note: You cannot stripe across multiple device types.

The creation of real-time filesystems requires the use of logical volumes. Logical volumes also provide the mechanisms for data striping.

For information on configuring and using RAIDs by creating logical units (LUN), refer to *IRIX Admin: Disks and Filesystems.*

This chapter contains:

- "Overview of Real-Time Filesystems" on page 20
- "Creating a Logical Volume Using XLV" on page 20
- "Choosing Parameters for Creating and Mounting an XFS Filesystem" on page 24
- "Using GRIO" on page 24

Overview of Real-Time Filesystems

A real-time filesystem is a subvolume in an XFS filesystem. When using an XFS filesystem as part of the VST clip cache, the real-time subvolume, if available, is where the media data is stored. Only real-time filesystems guarantee real-time I/O rates (GRIO).

Adding Data to a Real-Time Filesystem

Data can be read normally from a real-time filesystem, but standard utilities, such as *ftp* and *cp*, cannot be used to write the data. To add media to the real-time filesystem, use a customized version of ftp installed from *vcp_recorder_eoe.sw.ftpd*.

For more information, see *Video Server Toolkit Developer's Guide.*

Creating a Logical Volume Using XLV

A logical volume is a series of disks or RAID LUN that are treated as one disk. Logical volumes (XLVs) allow for the creation of a real-time subvolume to be used within a real-time filesystem. XLVs are also used to stripe data across a series of disks or LUN to form a subvolume.

Size of Volumes

Table 2-1 can help you determine the size of the volumes and therefore the amount of disk space you need according to different compression formats.

Table 2-1   Disk Space Requirements

Format	Rate (MB/sec)	Disk Space for One Minute (GB)	Disk Space for One Hour (GB)
	525/ 625	525/ 625	525/ 625
Uncompressed	20.6	1.26	75.6
	20.3	1,24	74.5
Rice Lossless	10.3	0.6	35.0
	10.2	0.59	34.6
DVCPRO	3.4	0.24	14.4
	3.4	0.24	14.4

Rice values are one half of uncompressed values.

The bitrate for MPEG clips depends upon the settings of the MPEG encoder which recorded it.

Logical Volume Creation Guidelines

When creating a logical volume to be used as part of VST's clip cache, the following guidelines should be observed:

- The data subvolume size should be 200 MB.

- You should create a real-time subvolume using a stripe size based on the striping guidelines in "Striping Volumes" on page 22.

- If GRIO is used, put only real-time subvolumes on all partitions of any disks or LUNS.

  Make sure that each disk or LUN is connected to a controller whose disks or LUNS are all used exclusively for real-time subvolumes.

- If a real-time subvolume is placed on a RAID subsystem(s), the data subvolume should be plexed on a pair of internal disks to improve reliability.

Striping Volumes

Part of the job of creating a logical volume is to supply XLV with values for striping the disks. Striping is the process of distributing media data across several disks so that different parts of the media can be simultaneously accessed, which provides higher, overall bandwidth.

Striping is highly recommended for real-time performance. You can stripe disks for the real-time subvolume of real-time filesystems (recommended), or for non-real time filesystems (not recommended).

For more information about XLV, see *IRIX Admin: Disks and Filesystems*, and Appendix A, "Example Filesystem and Plexing Setup."

Using Non-Real-Time, Non-Striped Filesystems

If using a non-real time filesystem without striping, add an entry to */usr/vtr/config/vtrfsinfo.conf* to specify the optimal I/O size for the disk device. For example, to specify an optimal I/O size of 4 MB for a single RAID device, use:

```
/dev/dsk/dks2d1s7 4m
```

Striping Guidelines

To stripe a volume, use the following guidelines:

- Choose a larger stripe-unit to improve performance, but do not exceed the optimal I/O size for the individual disk devices being striped. Exceeding the optimal size does not improve performance.

    – For the 9 GB and 18 GB drives that Silicon Graphics OEMs from IBM, the stripe-unit should be at least 256K, preferably 512K per disk drive.

    – For a fibre channel 4+1 RAID, the stripe-unit should be at least 1 MB, preferably 2 MB.

    – For a fibre channel 8+1 RAID, the stripe-unit should be at least 2 MB, preferably 4 MB.

- If a real-time subvolume is being striped, the real-time extent size of the XFS filesystem must match the stripe size given by the following formula:

    num-striped-disks `* stripe-unit * 512bytes/block`

Uncompressed Media Striping Guidelines

A stripe unit value of 552 disk blocks (226 KB) 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 filesystem block size.

A stripe unit value of 624 disk blocks (312 KB) 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 16 KB to match the recommended filesystem block size.

Choosing Parameters for Creating and Mounting an XFS Filesystem

When creating an XFS filesystem, use the following guidelines:

- The filesystem block size should always be 16 KB.

- The real-time extent size should be set to the optimal I/O size for the real-time subvolume.

  If the subvolume is striped, it must match the stripe size. If the subvolume is not striped (single disk or LUN), it should match the optimal I/O size for the disk system.

  For a single RAID, use 2 to 4 MB for the extent size. For more information about determining the optimal I/O size for a disk, see the striping guidelines in "Using GRIO" on page 24.

For more information about XFS, see *IRIX Admin: Disks and Filesystems*, and Appendix A, "Example Filesystem and Plexing Setup."

Using GRIO

High performance media streaming is of utmost importance in using VST. Guaranteed Rate I/O (GRIO) provides high performance streaming.

GRIO enables a user application to reserve part of a system's I/O resources for its exclusive use. It is used for "real-time" retrieval and storage of data streams. GRIO is an installable component in the IRIX operating system.

VST assumes that GRIO is running on real-time subvolumes. If it is not, aberrant behavior sometimes results; for example, record and play do not work while using an edit controller with Rice encoded clips.

VST also assumes that the disks are striped according to the guidelines specified in "Striping Volumes" on page 22.

When setting GRIO parameters, make the per-spindle I/O size at least 256K and preferably 512K.

For more information about GRIO and its installation, see *IRIX Admin: Disks and Filesystems*, and Appendix A, "Example Filesystem and Plexing Setup."

Adjusting the Maximum Bandwidth

To use GRIO, you must adjust the maximum bandwidth rating for the controllers and the disks by adding REPLACE or ADD lines to the */etc/grio_disks* file; REPLACE is for controllers, ADD is for disks.

Adjust the Maximum Bandwidth of the SCSI Controllers

To specify the bandwidth for the fibrechannel SCSI controller, use line like the following:

```
REPLACE /hw/module/1/slot/io7/MotherBoard/pci/3/scsi_ctlr/0 2048K 40
REPLACE /hw/module/1/slot/io7/MotherBoard/pci/1/scsi_ctlr/0 2048K 40
```

The fibrechannel SCSI controller is thus capable of 80 MB/second each, or 40 I/Os of 2 MB/second each. (80 MB/second is the recommended value to use.)

For each machine, the hardware tree path before the REPLACE lines must point to the fibrechannel cards. The example REPLACE lines point to PCI slots 3 and 1, respectively. The slot in which a card is installed can be determined by physical inspection.

Adjust the Maximum Bandwidth of the Disks

For example if you have a Ciprico 7000 fibrechannel RAID, you might use the following line:

```
ADD     "Ciprico Rimfire 7010   03.1" 2048K 22
```

This line says that the RAID should produce 44 MB/second, or, in this case, 22 I/O's at 2 MB/second. (44 MB/second is the recommended value to use.)

The I/O size should be the same as the stripe unit size of the filesystem on that disk. In this case it's 2Meg. You can have multiple ADD entries if there are filesystems with different stripe unit size on the same type of disk.

Changing the SCSI Identifier

The SCSI identifier in the ADD line must match the SCSI identifier of the attached RAID. If the identifiers do not match, GRIO will not work; playout and recording from that disk will be impossible.

To identify the correct SCSI ID of your attached RAID or other SCSI disk, use the following procedure:

To use GRIO, use the following procedure:

1. Use the *versions* command to make sure that the XFS GRIO subsystem, *eoe.sw.xfsrt*, is installed.

   ```
   % versions eoe.sw.xfsrt
   I  eoe.sw.xfsrt 02/24/97  XFS Realtime & Guaranteed-Rate Support
   ```

2. Use the hinv command to produce a list of the disks in the system.:

   ```
   # hinv -c disk
   Integral SCSI controller 3: Version Fibre Channel AIC-1160,
   revision 2
     Disk drive: unit 1 on SCSI controller 3
   Integral SCSI controller 2: Version Fibre Channel AIC-1160,
   revision 2
     Disk drive: unit 1 on SCSI controller 2
   Integral SCSI controller 0: Version QL1040B (rev. 2), single ended
     Disk drive: unit 1 on SCSI controller 0
     Disk drive: unit 2 on SCSI controller 0
     Disk drive: unit 3 on SCSI controller 0
     Disk drive: unit 4 on SCSI controller 0
     Disk drive: unit 5 on SCSI controller 0
   Integral SCSI controller 1: Version QL1040B (rev. 2), single ended
   ```

3. Using the information in the previous step, get the SCSI ID with the following command:

```
% fx `dksc(c,n)'
```

where *c* is the SCSI controller number, and *n* is the disk number on that controller.

For example, the root disk is controller 0 disk 1. An attached RAID should have its own controller and be the only disk on that controller.

In the following example, controller 2 is a fibrechannel controller and disk 1 is the RAID (the only disk on the fibrechannel card). The drive type is "Ciprico Rimfire 7010   03.1.".

```
# fx `dksc(2,1)'
Unable to determine drive type, defaulting to SCSI, ctrlr 0, drive 1
fx version 6.4, Sep 17, 1997
...opening dksc(2,1,0)
 fx: partition already in use by xlv logical volume
fx: devname          seq owner    state
fx: /dev/rdsk/dks2d1s7    1 xlv      part of xlv vol

fx: Warning:  this disk appears to have mounted filesystems.
        Don't do anything destructive, unless you are sure
        nothing is really mounted on this disk.
...drive selftest...OK
Scsi drive type == Ciprico Rimfire 7010    03.1

----- please choose one (? for help, .. to quit this menu)-----
[exi]t               [d]ebug/              [l]abel/
[b]adblock/          [exe]rcise/           [r]epartition/
fx> exit
```

4. Restart the *ggd* daemon:

```
# /etc/init.d/grio stop
# /etc/init.d/grio start
```

Now the user application can be started. Files created on the real-time subvolume volume can be accessed using guaranteed-rate I/O.

Disabling GRIO

Although using GRIO is recommended, VST enables you to disable it using the following control in */usr/vtr/config/system-defaults*:

```
vtr.storage.fs.grio.enabled false
```

Installing and Configuring Video Server Toolkit

This part of the book explains how to install and configure Video Server Toolkit.

Chapter 3, "Installing Video Server Toolkit," provides a step-by-step description of installing the VST software.

Chapter 4, "Configuring Video Server Toolkit," describes the configuration files and settings for VST.

Chapter 5, "Time Code Readers," provides a very brief discussion of time code and how to configure VST to work with the time code readers.

Chapter 7, "Configuring and Using the Sony or P2 Protocol," describes how to configure VST to use the Sony protocol.

Chapter 8, "Diaquest," describes how the VST can be used with Diaquest software to control an external media device.

Chapter 9, "Installing the Louth Automation Controller," describes how to configure the Louth automation controller to work with VST.

Chapter 10, "Configuring and Using V-LAN Transmitters," describes how to integrate V-LAN with VST.

Chapter 11, "Installing the Vela Research MPEG-2 Decoder," describes how to install and configure the Vela Research decoder.

Chapter 12, "Installing the DVB-ASI Adapter Board," describes how to install and configure the DVB-ASI encoder/decoder board.

Installing Video Server Toolkit

This chapter provides a step-by-step description of installing the Video Server Toolkit (VST) software in the following sections:

- "Pre-Installation Steps" on page 31
- "Installing VST" on page 33
- "Confirming the Installation" on page 36.
- "Plexing Your System" on page 39.

Pre-Installation Steps

Before you install VST, 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 VST requirements:

- "Collecting Required CDs" on page 32
- "Reading the Release Notes" on page 32
- "Installing Patches" on page 32
- "Verifying Your Network Connection" on page 33

Collecting Required CDs

To install VST, you need the following CDs:

- Video Server Toolkit
- IRIX
- Patches, for any patches required for the operation of VST not already included on the distribution CD. See the release notes.

  To meet certain VST prerequisites and to improve overall system stability, you must install the recommended patchset for your platform. The recommended patchsets are enumerated in the release notes.

  Caution: VST was tested for compatibility with specific hardware and software configurations, including the recommended patchset available at the time VST was released. Refer to the release notes for the supported configurations for this release.

Reading the Release Notes

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

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

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

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

Installing Patches

Before installing VST, install the IRIX patches required for VST, as specified in the Release Notes.

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

Warning: **After repartitioning the disk and installing IRIX, you must install the current, recommended patches before rebooting the system.**

Verifying Your Network Connection

To install VST over a network, you must be successfully connected to a network.

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

Updating Your Netscape Browser

The vcpmaint applet requires Netscape Communicator version 4.05 or higher. To download the latest version of this browser, use Netscape's Web page at *http://www.netscape.com*.

Installing VST

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

1. Become a superuser by entering the following command:

    ```
    % su
    ```

2. For a clean installation, remove any previous versions of *vcp_recorder_eoe*:

    ```
    # versions remove vcp_recorder_eoe
    ```

    This command cleans up all crontab entries.

    If you choose not to remove previous versions, the rotatelog installation exitop adds the rotatelog entry in crontab only if it does not exist already. This choice preserves user modifications.

3. Start the installation program and identify the source of the VST image, as follows:

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

For example, if you are installing over the network, use a line similar to the following:

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

Or, if you are installing from a CD, use

```
# inst -f cdHost:/CDROM/dist
```

Or, if you are installing from a local CD, use

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

This installation works for all operating systems.

4. List the subsystems to install as follows:

```
Inst> l
```

This command lists all the subsytems to install; Table 3-1 describes some of them.

Table 3-1   VST Default and Optional Subsystems

If Using	Required Subsystems	Description
Non-HTML documentation	vcp_recorder_eoe.books.Admin Guide	Installation and Administration Guide
Non-HTML documentation	vcp_recorder_eoe.books.DevelG uide	Developers Guide
HTML documentation	vcp_recorder_eoe.books.Admin GuideHTML	HTML Installation and Administration Guide
HTML documentation	vcp_recorder_eoe.books.DevelG uideHTML	HTML Developers Guide
Man pages	vcp_recorder_eoe.man	Man pages
Base man pages	vcp_recorder_eoe.man.base	Base man pages
Tools man pages	vcp_recorder_eoe.man.tools	Base applications man pages
Real-time file system	vcp_recorder_eoe.man.ftpd	Man pages for enhanced FTP daemon
Release notes	vcp_recorder_eoe.man.relnotes	Release notes

Table 3-1 (continued)   VST Default and Optional Subsystems

If Using	Required Subsystems	Description
Base software	vcp_recorder_eoe.sw.base	VST Base software.
DiaQuest	vcp_recorder_eoe.sw.diaquest	DiaQuest deck-control module
DIVO	vcp_recorder_eoe.sw.divo	DIVO video module
DVB-ASI	vcp_recorder_eoe.sw.dvb-asi	Required for ViewGraphics DVB/MPEG mediapump module
Adding clips to system	vcp_recorder_eoe.sw.fsmon	File System Import Monitor (FSmon)
FTP required for real-time systems	vcp_recorder_eoe.sw.ftpd	Customized FTP daemon necessary for recording to real-time file systems.
Horita time code reader	vcp_recorder_eoe.sw.horita	Horita time code input module
Little Red time code reader	vcp_recorder_eoe.sw.little-red	Miranda Little Red time code input module
Automation controller using Louth protocol	vcp_recorder_eoe.sw.louth	Louth VDCP control module
Archiving	vcp_recorder_eoe.sw.studiocentral	StudioCentral 2.0 archive interface
MPEG decoding using Vela or DVB-ASI	vcp_recorder_eoe.sw.mpeg	MPEG2 tools
O2 workstation	vcp_recorder_eoe.sw.o2video	O2Video module
Sony or Odetics protocols	vcp_recorder_eoe.sw.sony-odetics	Sony/Odetics control module
Operator tools	vcp_recorder_eoe.sw.tools	VST UI applications.
Vela decoder	vcp_recorder_eoe.sw.vela	VST Vela SCSI MPEG2-decoder module
VLAN transmitter	vcp_recorder_eoe.sw.vlan	VLAN deck-control interface

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

Note: If you installed the drivers for hardware, such as a VELA, 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.

5. Mark for installation any optional VST related software you might need. For example, if you are using VELA, MPEG, Louth, DVB-ASI, Sony protocol, or Odetics protocol, install one or more of the following options, as appropriate:

```
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
Inst> i vcp_recorder_eoe.sw.sony-odetics
Inst> i vcp_recorder_eoe.sw.ftpd
Inst> i vcp_recorder_eoe.sw.dvb-asi
```

6. Initiate the installation, as follows:

```
Inst> go
```

7. Quit the installation program, as follows:

```
Inst> quit
```

8. To run a secondary server that takes over VST operations when the primary server fails, you must repeat the above installation procedure for the secondary server and install IRIS FailSafe, as described in Chapter 6, "Installing IRIS FailSafe."

Confirming the Installation

To confirm the successful installation of VST, complete the following steps:

1. Start VST by entering the following command as a superuser:

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

If VST starts, on an O2 workstation bitmapped console display only, the VST window is displayed.

2. To see if VST started, run

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

If VST is running, the command responds:

```
Video Server Toolkit on hostname is running.
```

3. Verify the media ports are 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
```

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

```
# cd /usr/vtr
# ls
bin     clips   config   data     images   index    install
   lib32    lib
# cd /var/adm/vtr
crash   logs
# cd /var/adm/vtr/logs
# ls
vtrlog
```

vtrlog appears only after VTR-Recorder has been started.

The crash directory contains crash files for *vvtr*.

Table 3-2 describes these directories.:

Table 3-2   /usr/vtr Subdirectories

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

Common DSOs

Table 3-3 lists the dynamic shared objects (DSOs) installed in *usr/vtr/lib32*:.

Table 3-3   Common DSOs

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

Plexing Your System

After installing the VST images, you can plex your system to increase data output. For a detailed example of plexing a system, see Appendix A, "Example Filesystem and Plexing Setup."

Configuring Video Server Toolkit

Video Server Toolkit (VST) uses the following files to configure VST so that it works with all the hardware and software in the system:

- */usr/vtr/config/control-in.conf,* which configures VST to be controlled by remote devices. Use this file to configure VCP to work with controllers compatible with Sony, P2, Odetics, and Louth protocols, and timecode reader devices

- */usr/vtr/config/control-out.conf,* which configures VST to control other decks. Use this file to configure VST to control VTRs using VLAN and Diaquest software.

- */usr/vtr/config/device-defaults,* which configures audio and video devices. To specify system wide behavior, use */usr/vtr/config/system-defaults.*

- */usr/vtr/config/system-defaults,* which contains vcp controls that specify system behavior, as opposed to unit-specific behavior, which is specified by */usr/vtr/config/device-defaults,*

- */usr/vtr/config/studiocentral.conf,* which configures the archival system.

- */usr/vtr/config/ports.conf,* which maps physical port names, such as *DIVO_0,* to logical port names, such as *master_1.*

- */usr/vtr/config/vtrd.conf,* which is the VTR daemon configuration file.

- */etc/ioconfig.conf,* which assigns names to PCI slots.

- */usr/vtr/config/vtr_ioconfig.conf,* which contains configuration information for the Vela decoder. For more information, see "Configuring the Vela Decoder" on page 86.

- *vtrfsinfo.conf,* which defines I/O size and alignment size for none real-time filesystems.

This chapter describes how to configure VST in the following sections:

Configuring the Device Interface

Device files contain variables, called controls, that influence the behavior of specific devices. There can be one device file per device in your system.

Naming Specific Devices in Your System

The directory, */usr/vtr/config/devices_defaults,* contains files that correspond to the media devices used by the VST; there is one file for each device or device type. These files define machine-specific behavior.

Valid file names in *devices_defaults* include:

- MEDIA, for behavior common to all connected media devices.

- mvp, for an O2 workstation. Because an O2 workstation can only take one analog or digital video card, there is no need to use other mvp names, such as mvp_0.

- divo, for all non-DVB-ASI-capable DIVO boards in a system, or divo_0, divo_1, divo_2, and so on to name a specific DIVO board is in use in a VST server.

- DIVO_DVC, for DVCPRO boards.

- vela, for all Vela devices connected to a system, or vela_0, vela_1, vela_2, and so on. if more than one VELA device is in use in a VST server.

- vlan, for all V-LAN devices in a system, or vlan_0, vlan_0_1, vlan_0_2, and so on. if there is more than one V-LAN transmitter connected to a VST server.

- dq, or dq-0, dq_1, dq_2, and so on. when VST is using Diaquest software to control external VTRs.

- dvb-asi, for media devices connected to a DVB-ASI adaptor board, or dvb-asi_0, dv-asi_1, dvb-asi_2, and so on. if there is more than one DVB-ASI adaptor board connected to a VST server.

Setting Device Parameters

You use VST controls to specify unit-specific behavior. For example, for the first DIVO_DVC device, you would find in *device-defaults* directory a file, called *DIVO_DVC_0*. In it, you might find:

```
# Format
vtr.media.video.input.compression.type dvcpro
vtr.media.clip.format movie/dif

# Audio
vtr.media.audio.input.port  ""
vtr.media.audio.output.port ""
```

Configuring System Defaults

You use VST controls in a file in the */usr/vtr/config/system-defaults* directory to specify default system-wide behavior. For example, a file in the *system-defaults* directory might contain:

```
vtr.main.thread.cpu_limit 50
vtr.main.thread.cpu_limit_interval 5
vtr.main.timing_standard 625
```

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

Configuring DVCPRO

If VST will play 625/50 clips transferred from a Panasonic NewsBYTE, the following control must be set in the */usr/vtr/config/system-defaults/main* configuration file:

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

If the default timing standard is set to 625, 525/60 clips transferred from a NewsBYTE will not play on VST because it assumes they are 625/50.

Other Controls to Check

Check the following controls depending on your hardware configuration:

vtr.media.dvb-asi.port
                    indicates which board the output goes through

vtr.media.dbv-asi.seamless_playback
                    splicing turned on when this control is set to TRUE

vtr.media.dbv-asi.halt_on_underflow
                    should be set to false

vtr.media.mpeg.pid.video
                    set the PID of the output video

vtr.media.mpeg.pid.pmt

>set this control to the output program map table PID. The PIDs should be unique for each stream. The default for audio is video+1;make sure PMT is not video+1.

vtr.media.mpeg.program_number

>set this control to a unique for each stream

vtr.media.dvb-asi.port.total_bitrate

>set this control to the total bit rate of the decoder, for example, you might set this control to 52000000 for the Tandberg decoder.

Configuring Audio Support

For the Vela decoder output, the decoder includes audio outputs that are decoded from the MPEG2 stream. There are no VST controls for adjusting the audio outputs. The audio configuration is managed by the decoder.

For the DVB and ATM outputs, the MPEG2 stream is transmitted in its compressed form, including the compressed audio.

For the DIVO and O2 outputs, a unit may select the audio port for:

- audio output during playback

- audio input during recording

The controls that configure this are *vtr.media.audio.output.port* and *vtr.media.audio.input.port*, respectively.

The default settings for the ports are DefaultOut and DefaultIn, respectively. The default audio input and output ports are selected using the IRIX Audiopanel control panel.

For more information about audio ports, see the vcp-recorder-controls(5) man page.

Audio Ports

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 "LineIn," or "Microphone," and applies only 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"
```

Port Comparisons

The following list is an example of video, audio, and serial port configurations on different systems:

- Origin200 GIGAchannel has 3+4 PCI slots and 5 XIO slots; one XIO slot can be used for 100BaseT XIO, which has 6 serial ports

- Origin2000 (one node; two CPUs) has 3 PCI and 4 XIO slots

- Origin2000 (two nodes; four CPUs) has 3 PCI and 10 XIO slots

Configuring the Audio Ports

To identify which audio port to use, configure a unit in */usr/vtr/config/device-defaults/**. For example, in DIVO_1 you would enter:

```
vtr.media.audio.input.port RAD1.AESIn
vtr.media.audio.output.port RAD1.AESOut
```

Alternately, use the SET command in a mvcp control connection for a unit, as follows:

```
SET <unit> MED vtr.media.audio.input.port RAD1.AESIn
SET <unit> MED vtr.media.audio.output.port RAD1.AESOut
```

SET must be issued for every unit.

When using SET, values are not kept after a crash, reboot, or manual restart.

Disabling Audio

vvtr can malfunction or lose sync if there is no input audio. If your media does not have audio, do the following:

1. Turn the audio off: in the */usr/vtr/config/device-defaults* directory, edit the DIVO file and add:

   ```
   vtr.media.audio.input.channels.clip "0"
   ```

2. If you do not have an audio source, disable input audio by setting the unit control:

   ```
   vtr.media.audio.input.port ""
   ```

control-in.conf File

Use the */usr/vtr/config/control-in.conf* file to configure VST so that it can be controlled by remote devices. Currently-supported remote devices are those supporting the protocols defined by Sony, Louth, and Odetics.

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

- Serial port configuration—one line per device
- Other control ports—one line per device
- Signal configuration—one line per device

Serial Port Configuration Lines

You use the configuration file, */usr/vtr/config/control-in.conf*, to specify basic connection parameters between the controlling devices (Sony, Louth, and Odetics) and VST. Connection parameters include such information as which serial port controls which video board and how.

For each board controlled by VST using the Sony protocol, you need a line in */usr/vtr/config/control-in.conf* that uses the following format:

```
protocol_name   connection serial_port    baud_rate   parity
    signal_number     frame_rate frames_before_activation
```

The following line shows a sample configuration:

```
sony    rs422   4    38400    1    1    29.97    0
```

In this configuration, the VST controls a media device at 29.97 frames per second, using the Sony protocol and serial port 4 running at 38400, odd parity, with a zero-frame command preroll. The preroll should always be zero, except for Louth controllers in which case the preroll should be three.

Note: Two serial port connections are required if you want to play and record simultaneously. In this case, you must add two lines to the */usr/vtr/config/control-in.conf* file; one for each port. Currently, this option is not supported by the Sony protocol.

The following sections describe each variable in this configuration line.

protocol_name

The valid values of *protocol_name* include:

- sony

- p2

- odetics

- louth

- mvcp

- hsip

- littlered

hsip is used for the Horita time code reader.

Use the value that corresponds to the protocol used by the device you are configuring to work with VST.

connection

The valid values of *connection* include:

- rs232, for a RS-232 control connection.

- rs422, for a RS-422 control connection.

- tcp, for TCP/IP

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

Note: rs422 is not supported on O2.

serial_port

The values of *serial_port* must be an integer that corresponds to the serial port on the VST server where the control video device is connected. On the Origin2000, for example, there are two serial ports. A value of 4 for *serial_port* means that a device is connected to the */dev/ttyd4* port on the Origin2000.

Note: TCP/IP ports and signal/media ports are unrelated.

baud_rate

The *baud_rate* value specifies the connection speed between the serial port on the server and the video device. Valid values include 14400, 28800, and 38400. For Sony and Odetics devices, this value is always 38400.

parity

- *parity* is an optional setting; it can have one of the following values:
    - 0, indicating no parity
    - 1, indicating odd parity
    - 2, indicating even parity

Odd parity is used by most controllers. Currently, odd parity is the only supported option.

signal_number

The *signal_number* value is the number that corresponds to the signal number associated with a video device. For more information, see "Signal Configuration Line" on page 52.

frame_rate

The *frame_rate* value specifies the number of frames per second displayed by the controlled video device. Valid values include:

- 24, for film

- 25, for the PAL video format

- 29.97, for NTSC drop-frame video

- 30, for NTSC non-drop-frame video

frames_before_activation

When VCP_Recorder receives a command, only the next frame to enter the frame buffer can be affected by the command. For example, if the frame buffer 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 frame buffer are displayed; the third is not displayed, however, because it is affected by the stop command.

You can modify the number of frames that display before a command begins by setting the *frames_before_activation* value. Valid values are zero or positive integers.

The minimum value equals the number of frames buffered in your system (for example, 3) because those frames are already on their way out of the frame buffer when the command arrives. You can define the maximum number of frames using the following control:

```
vtr.media.video.output.max_queued_frames
```

A small number for *frames_before_activation* 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, Pause.

Note: For more information about setting controls, see the *Video Server Toolkit Developer's Guide.*

frames_before_activation allows you to account for system latencies and helps produce frame-accurate behavior. System latency is typically a result of frame buffers or a slow network. If, for example, a controller always sends commands late by one frame, you can decrease the *frames_before_activation* value by one to account for it.

A value of zero for *frames_before_activation* means to begin as soon as possible.

Some commands in the Sony and Odetics protocols imply a pre-defined frame delay. *frames_before_activation* does not affect these commands.

Note: If VST is stopped and then receives a PLAY command, and *frames_before_activation* is set to 3, the first frame displays on the third "frame time" after the command is received. Even though video is not playing the first two "frame times," there still is a two frame delay in the execution of the PLAY command.

Signal Configuration Line

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

```
signal 1 DIVO_4
```

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

Network Transmission Configuration Line

The VST supports MVCP, Sony, P2, Odetics, and Louth device control protocols. MVCP is treated differently for two reasons:

- MVCP is used over TCP/IP and not over the serial port. Consequently, much of the information provided for the other video control protocols is not applicable.

- A single mvcp connection can control all of the available media (signal) ports on a machine, unlike the other protocols, which need to be assigned a media port. Therefore, media port-related information is also unneeded.

To configure mvcp, you must enter a configuration line in *usr/vtr/config/control-in.conf* to specify the network transmission used by VST, in the following form:

```
protocol_name     transmission_medium     port_number
```

For example, the configuration line

```
mvcp    tcp    5250
```

specifies that the mvcp protocol uses the tcp transmission medium on port number 5250.

Warning: **This configuration line must be present for VST's GUIs to work.**

The following sections describe each variable in this configuration line.

protocol_name

The only valid value of *protocol_name* is mvcp.

transmission_medium

transmission_medium specifies the manner in which data is transmitted over a network. This value can only be set to tcp.

port_number

port_number is the port number an application uses to connect over the network. The value 5250 is the standard, but *port_number* can be set to any unreserved port number above 1024. There is no reason to use a number other than 5250 unless another application running on the server also wants to use that port number for something else.

Note: TCP/IP ports and signal/media ports are unrelated.

Example control-in File

Here is a sample *control-in.conf* file:

```
# This defines the normal VCP-Recorder MVCP TCP port

mvcp tcp 5250

# Other control ports

#sony   rs422 2 38400 1 1 29.97 3
louth   rs422 2 38400 1 1 29.97 3
louth   rs422 5 38400 1 2 29.97 3
#
louth   rs422 6 38400 1 3 29.97 3
louth   rs422 7 38400 1 4 29.97 3
#louth   rs422 8 38400 1 5 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:        VCP-Recorder port name (e.g., DIVO_1)

signal 1 DIVO_DVC_0
signal 2 DIVO_DVC_1
signal 3 DIVO_DVC_2
signal 4 DIVO_DVC_3
#signal 5 DIVO_4
#signal 2 DIVO_3
```

control-out.conf File

Use the */usr/vtr/config/control-out.conf* file to configure VST so that it can control other VTRs.

Use the configuration file, */usr/vtr/config/control-out.conf*, to specify basic connection parameters between the Sony/P2 VTR and VST. Connection parameters include such information as which serial port controls which VTR and how.

For each VTR controlled by VST using the Sony protocol, you need a line in */usr/vtr/config/control-out.conf* that uses the following format:

```
protocol_name connection serial_port baud_rate parity
```

The following line shows a sample configuration:

```
sony    rs422    4    38400    1
```

The line has the same meaning when substituting "p2" for "sony."

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/control-out.conf* file, one for each port.

The following sections describe each variable in this configuration line.

protocol_name

The valid values of *protocol_name* include:

- sony
- p2
- vlan

Note: sony is equivalent to p2.

Use the value that corresponds to the device you are configuring to work with VST.

connection

The valid values of *connection* include:

- rs232, for a RS-232 control connection
- rs422, for a RS-422 control connection

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

Note: rs422 is not supported on O2.

serial_port

The value of *serial_port* must be an integer that corresponds to the serial port on the VST server where the control video device is connected. On the Origin2000, for example, there are four serial ports. A value of 4 for *serial_port* means that a video device is connected to */dev/ttyd4* port on the Origin2000.

The RS-422 pinouts are non-standard; see Appendix D, "RS-422 Pinouts."

baud_rate

The *baud_rate* value specifies the connection speed between the serial port on the server and the video device. Valid values include 14400, 28800, 38400, and 56000. Sony and Odetic devices must use 38400.

parity

- *parity* is an optional setting; it can have one of the following values:
  - 0, indicating no parity
  - 1, indicating odd parity
  - 2, indicating even parity

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

Defining the DMA

The maximum 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 4MB which for most VST installations is too small.

To increase the system's maximum DMA size (to 10MB, in this example), use the following procedure:

1. Start systune to change the DMA size:

   ```
   # systune -i
   ```

   The system responds:

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

   systune->
   ```

2. Enter the new DMA size.

   ```
   systune-> maxdmasz 10000000
   ```

   The system responds:

   ```
         maxdmasz = 4000000
      Do you really want to change maxdmasz to 10000000? (y/n)
   ```

3. Enter **y.** for yes.

   The system responds:

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

4. Quit systune:

   ```
   systune-> quit
   ```

5. Reboot the machine.

   ```
   # reboot
   ```

57

Configuring the Archival System

VST works with archive systems, such as StudioCentral 2.0 (SC 2.0), to archive media. The VST library that communicates with SC 2.0 is installed by default with the installation of VST. To record media into SC 2.0, however, you must install the optional software, *vcp_recorder_eoe.sw.ftpd*, which is a customized version of FTP that can write to a real-time filesystem.

For the archival system to work, StudioCentral 2.0, Asset Transfer Service (ATS) services, and the servers containing the media content must be running on a networked server.

Note: To run the ATS services and content servers, other StudioCentral services must be running. Please consult the StudioCentral 2.0 documentation for further details.

For a general description of the archive and using it, see the *Video Server Toolkit Developer's Guide.* It provides an overview of the VST server, ATS server, and the archive server.

Editing the Configuration File

Each line in the archive configuration file, */usr/vtr/config/studiocentral.conf,* has the following syntax:

ATS_Server_Name   *ATS_Service_Name*   [shared=*archive_Server*:*Path_To_Media*]

ATS_Service_Name can also be the port number of an ATS service.

If any archive server runs on the same filesystem as any of the VST caches, you can specify the path to the media using the *shared* directive so the media is not copied but linked.

Note: The association between the ATS server and its archive servers is specified in the configuration files of StudioCentral 2.0.

For example, the following lines configure VST to work with two archives:

```
myATSMachine1 ats
myATSMachine2 1590 shared=myContentServer:/ATScontent/vol1
```

In this example, *myATSMachine1* and *myATSMachine2* are the names of the ATS servers, ats and 1590 are the service name and the port number of the ATS services, respectively, and */ATScontent/vo*l is the path to the media stored by ATS's content server, named *myContentServer,* on the machine running VST

The archive server accessed by *myATSMachine1* is not the same as the machine running VST.

Configuring Automatic Startup at Boot Time

By default, when VST is installed, it does not enable itself to be automatically started when the system boots. If you are a superuser, you can configure VST to start automatically at boot time using *chkconfig*. VST is controlled by the setting of the vtr option:

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

Setting Video Output Sync Source

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

vtr.media.video.output.sync_source may be set for mvp and all DIVO boards in the server in the *device-defaults/DIVO* or */mvp* file, or may be set per board in the *device-defaults/DIVO_n* or *mvp_n* files.

Setting the Compression Type

By default, VST uses Rice packing when recording clips using a DIVO and JPEG compression when recording clips on an O2 workstation.

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

On DIVO_DVCPRO, set *vtr.media.video.input.compression.type* to DVCPRO to default to DVCPRO compression.

Mapping Physical Ports to Logical Port Names

You use the */usr/vtr/ports.conf* file to map physical port names, such as *DIVO_0*, to logical port names, such as.

```
port DIVO_0 ChA "On-Air Channel A"
port DIVO_1 ChB "On-Air Channel B"
```

Start Up Options

vtrd.conf is the configuration file that vtrd reads when it starts. This is the VST "super-server," which starts child processes, such as *vvtr*, during startup and as needed.

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*.

You use the */usr/vtr/vtrd.conf* file to configure the VTR daemon. Each line in the files has the following format:

```
name cpu# priority min-retry(ms) max-retry(ms) term-signal command
```

For example;

```
vvtr * * * * TERM "/usr/vtr/bin/vvtr -l p"
vtrmaintd      * * * * TERM "/usr/vtr/bin/vtrmaintd"
```

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 4-1 describes the start up options for VST.

Table 4-1   Start Up Options

Option	Description
-v *loglevel*	Sets the severity level of the messages that are written to the console log.
	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.
	See on Table 14-2 on page 101 for an explanation of the VST 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.
	The following may be specified for this option:
	-l l, don't include the severity code in each log message -l t, don't include the timestamp in each log message -l p, don't include the process ID in each log message
	See "VST Log Messages" on page 104 for more information about the format of the log messages.
-p	Specifies that VST should not run at high priority (p). If this option is omitted, VST runs at the highest system priorities.
	VST 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 VST were to go into an infinite loop while running at the highest system priorities, it would lock out all other processes on that processor.)
	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 VST, 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 100.

Automatic DVB-ASI Configuration

/etc/ioconfig.conf is an automatically generated and maintained file. DVB-ASI devices show up as follows in this file:

```
0 /hw/module/1/slot/io7/xbox_dualxtown/pci/3/Dynamo
1 /hw/module/1/slot/io1/MotherBoard/pci/5/Dynamo
```

The first line says that DVB-ASI 0 is in PCI slot 3 of the GIGAchannel box. The second line says that DVB-ASI 1is in PCI slot 5 of Module #1 Motherboard.

Configuring for 625/50 Systems

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

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

This control sets the timing of the system, including the DIVO cards, to 625/50.

Make sure, in */usr/vtr/config/device-defaults/DIVO*, the control, *vtr.media.video.output.timing*, is set to its default value, *system*.

Configuring Non-Real-Time Filesystems

The *vtrfsinfo.conf* file sets the I/O size and alignment size for non-real-time filesystems.You must create this file with a text editor, such as vi, and use the following syntax:

device_name   I/O_size   alignment_size

For example:

```
/dev/xlv/striped 4m 4m
```

device_name

device_name is the device path for the disk or the XLV volume. Valid values are:

```
/dev/dsk/dks2d1s7
/dev/xlv/striped
```

I/O_size

I/O_size is the desired I/O size for the non-real-time filesystem. Make this value the same as the stripe size when you are using a striped XLV. If you are using a single disk, *I/O_size* should be 256K - 512K.

Some valid (integer) values are 4096k and 2m.

alignment_size

The *alignment_size* should always be equal the *I/O_size* unless you transfer the clips to a system with a different alignment size; in that case, use the alignment size of the other system

Some valid (integer) values are 4096k and 2m.

Configuring VST for Your Edit Controller

Because VST is a buffered system, there is a delay between a command and its execution. It is important to make equal the delay time in your edit controller with the delay time built into VST.

The two VST controls that set the VST delay time are:

```
vtr.control.edit.delay
vtr.control.ee.delay
```

The default values, which are the minimum delay values, are:

- Rice, uncompressed: 5 frames
- DVCPRO: 8 frames

If you set the values for these controls less than these default values, the lower values have no effect; the minimum, default value is used instead.

For more information about these controls, see, "Video Server Toolkit Controls."

Make sure to set your edit controller to the same values you use for these controls. To set the delay time in your edit controller, consult its user's manual.

Time Code Readers

Video Server Toolkit (VST) supports the Horita and the Miranda Little Red time code readers. These readers facilitate slaving media devices to the time of day.

This chapter provides a very brief discussion of time code and how to configure VST to work with the time code readers.

Overview of Time Code

Time code is used by videotape or digital decks to locate individual frames of video. For example, "In" and "Out" points mark the beginning and ending frames of sections to be played or recorded; the time code identifies those frames.

Time code is formatted as follows:

`hh:mm:ss:ff`

- *hh* stands for hours. It is base 24.

- *mm* stands for minutes. It is base 60 because they correspond to elapsed time in the video.

- *ss* stands for seconds. It is base 60 because they correspond to elapsed time in the video.

- *ff* stands for frames. It is base 30 because 30 frames display in one second in NTSC video or base 25 for PAL.

Setting the Time

VST uses the time-code input to set its internal "wall-clock" time. By default, it also sets the IRIX time. This default is controlled by the vtr.time.slave_system_time control in */usr/vtr/config/system-defaults/time*.

Configuring VST to Use Time Code Readers

VST supports the Horita and the Little Red time code readers. To configure VST to work with them, you must add a line for each device to *control-in.conf*, for example,

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

The format of the configuration line is explained in "control-in.conf File" on page 48.

Installing IRIS FailSafe

If you plan to use a dual server configuration so that a secondary server can take over if the primary server fails, you need to install IRIS FailSafe version 1.2 and patch 3422. The IRIS FailSafe installation is described in detail in *IRIS FailSafe Installation and Maintenance Instructions* and the *IRIS FailSafe Administrator's Guide*.

Note: Make certain that you review these guides before beginning the IRIS FailSafe installation.

Appendix A, "Example Filesystem and Plexing Setup" provides a complete example of a filesystem setup and a software installation.

This chapter describes the following:

- "The IRIS FailSafe Configuration" on page 67
- "IRIS FailSafe" on page 68

The IRIS FailSafe Configuration

IRIS FailSafe is a software product that allows a pair of servers to be used in a redundant configuration. The servers are configured with the Video Server Toolkit (VST) software and share the VST filesystems.

In an IRIS FailSafe configuration, the operating system on each server is configured for VST. The disks on each server store the operating system and the shared RAIDs store the clips and index filesystems. The RAID is shared by physically attaching it to an Emulex LH5000 Digital Fibre hub and connecting the servers to two other ports in this hub. The two servers along with the shared RAID(s) form an IRIS FailSafe cluster.

The servers also share a public IP alias or a name users can use to connect to the servers. If a server fails, the backup server takes over the shared RAIDs as well as this IP alias, which users still use to connect to the servers. To the user, a failover looks the same as a server that crashed and got rebooted very quickly.

The servers communicate using a special serial connection. When the backup server detects a problem with the active server, IRIS FailSafe unmounts the filesystem from the active server and automounts them on the secondary server. VST on the secondary server will detect this action and add the clips to its tables and be ready to play them.

IRIS FailSafe

The IRIS FailSafe configuration is implemented based on the Silicon Graphics IRIS FailSafe model using a shared disk in an active/backup cluster. To correctly install and configure IRIS FailSafe, follow these steps:

1. Install the IRIS FailSafe software from the IRIX distribution CD.

   ```
   # inst -f /CDROM/dist
   Inst> keep *
   Inst> install ha ha_www ha_fsconf
   Inst> go
   Inst> quit
   ```

2. Install the latest FailSafe patch from the patches CD.

   ```
   # inst -f /CDROM/patches/patch3422/dist
   Inst> install *
   Inst> go
   Inst> quit
   ```

3. Set the system variables on the primary server.

   ```
   # nvram AutoLoad Yes
   # nvram scsihostid 0
   # chkconfig failsafe off
   ```

4. Set the system variables on the secondary server.

   ```
   # nvram AutoLoad Yes
   # nvram scsihostid 8
   # chkconfig failsafe off
   ```

5. Configure the network interfaces on the primary server. Edit the */etc/hosts* file to include:

- a private network between the servers (not broadcast)

- hostname and address for each server

- a single hostname and address for the entire IRIS FailSafe configuration (also called the public IP address)

An example of the entries in the */etc/hosts* file is shown in Example 6-1. Your network administrator should provide you with private and public host names and an IP address. The two private IP addresses and the public IP address should be on the same subnet.

Example 6-1   Example /etc/hosts File

```
# This entry must be present or the system will not work.
128.64.0.1       localhost

# Primary FailSafe Host
192.70.0.100  fsprimary.engr.sgi.com fsprimary
192.0.3.1        priv-fsprimary.engr.sgi.com priv-fsprimary

# Secondary FailSafe Host
192.70.0.101  fssecondary.engr.sgi.com fssecondary
192.0.3.2        priv-fssecondary.engr.sgi.com priv-fssecondary

# FailSafe Alias for single system image
192.70.0.109 fssystem.engr.sgi.com fssystem
```

6. Place the same network information in the */etc/hosts* file on the secondary server.

7. Add the following entries to the primary server */etc/config/netif.options* files.

```
if1name=et0
if1addr=$HOSTNAME

if2name=et1
if2addr=priv-$HOSTNAME
```

8. Place the same information in the */etc/config/netif.options* files on the secondary server.

9. Turn routing off on the primary server by modifying the */etc/config/routed.options* file to contain the following:

```
-h -q
```

10. Place the same entry in the */etc/config/routed.options* file on the secondary server.

11. Reboot the primary server to enable the network changes.

    # **reboot**

12. Reboot the secondary server to enable the network changes.

    # **reboot**

13. Use the *ping* command from the secondary server to check the connection. For example:

    # **ping fsprimary.engr.sgi.com**

14. Use the *ping* command from the primary server to check the connection. For example:

    # **ping fssecondary.engr.sgi.com**

15. Set host resolution to local first in the */etc/resolve.conf* file on the primary server.

    hostresorder local nis bind

16. Set host resolution to local first in the */etc/resolve.conf* file on the secondary server using the same entry.

17. Place the following alias in the */etc/aliases* file to handle admin messaging from IRIS FailSafe on the primary server. Use the email address of the person who administers the system.

    fsafe_admin:postmaster
    postmaster:**fs_admin_person@another_system**

18. Run the *newaliases* command the enable the changes

    # **newaliases**

19. Place the same aliases in the */etc/aliases* file on the secondary server and use the *newaliases* command to enable the changes.

20. Edit the */etc/inittab* file on the primary server to reserve the private IRIS FailSafe serial connection between the servers. Make certain that the t2 entry is:

    t2:23:off:/sbin/getty -N ttyd2 co_9600   # port 2

21. Run the *init q* command to enable the serial port changes.

    # **init q**

22. Change the */etc/inittab* file t2 entry on the secondary server to match that of the primary server and then run the *init q* command to enable the changes.

23. Make sure you install the failsafe subsystem while installing VST:

```
# inst -f /CDROM/dist
Inst> install vcp_recorder_eoe.sw.failsafe
Inst> go
Inst> quit
```

24. Create the XLV logical volumes and XFS filesystems which are to be shared by following the instructions in the guide, *IRIX Admin: Disks and Filesystems.*

    Remember that each XLV logical volume must be owned by the node that is the same as the primary node for running VST. To simplify the management of the node names (or owners) of volumes on shared disks, follow these recommendations:

    • Work with the volumes on the shared disk/RAID from the primary node.

    • If you did not use the primary to create the volumes, you can change the node name to the primary node using *xlv_mgr*.

25. Copy */usr/vtr/failsafe/ha.conf* to */var/ha/ha.conf* onto the primary and secondary servers.

26. Copy */usr/vcr/failsafe/chkvcpr* to */var/ha/actions/chkvcpr* onto the primary and secondary servers.

27. If necessary, change the default mount point for the shared file systems in */var/ha/ha.conf* on the primary and secondary servers.

    By default, the shared filesystem is mounted at */usr/vtr/clips.* If you want to install a separate shared filesystem that stores the indices for the clips:

    • Add a new filesystem block just like the one in the sample file for filesystem *fcraid1.*

    • Add a new volume block, as shown for volume *fcraid1.*

    See the *IRIS FailSafe Administrator's Guide* for more details.

28. Change the default hostnames, the *fsprimary.engr.sgi.com* entry, in */var/ha/ha.conf* to the name of your primary server.

29. Change the *fssecondary.engr.sgi.com* entry in */var/ha/ha.conf* to the name of your secondary server.

30. Change the default IP address, 198.29.66.18, in */var/ha/ha.conf* to point to the IP address of your primary server.

31. Change the default IP address, 198.29.66.90 in */var/ha/ha.conf* to the IP address of your secondary server.

32. Change the default netmask values and the *broadcast-addr* variable to the correct values for your network on the primary and secondary servers.

    The *broadcast-addr* variable should point to the broadcast address of the IP subnet on which the servers are present. Both servers must be on the same IP subnet.

33. Perform the following steps as a superuser on the primary and secondary servers:

    ```
    # chown root.sys /var/ha/ha.conf
    # chmod 500 /var/ha/ha.conf
    # /usr/etc/ha_cfgchksum
    ```

34. Make sure the output of *ha_cfgchksum* is the same on both servers.

35. Make sure no errors are reported when you run on the primary and secondary servers:

    ```
    # /usr/etc/ha_cfgverify
    ```

36. Make sure the shared filesystems are not shown on the primary and secondary servers when you run *df* or in the file, */etc/fstab*.

37. Make sure the mount points are the same on both servers for the shared filesystems.

38. Execute the following command first on the primary server and then on the secondary server:

    ```
    chkconfig failsafe on
    /etc/init.d/failsafe start
    ```

Configuring and Using the Sony or P2 Protocol

The Sony/P2 protocol is partially supported by Video Server Toolkit (VST) so that VST can be controlled by standard VTR controllers. The configuration information makes VST 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 steps describe how you configure VST to work with the Sony protocol:

- "Modifying the Devices Configuration File" on page 73
- "Completing the Configuration" on page 75
- "Changing the Clip Loaded in a Sony Controlled Logical Unit" on page 75
- "Serial Cabling" on page 76
- "Configuring Your Edit Controller" on page 76

Note: The O2 workstation does not support deck emulation in version 1.1 of VST.

Modifying the Devices Configuration File

The directory, */usr/vtr/config/system_defaults*, contains files that correspond to the control ports or storage devices used by the system; there is one file for each device. The files define machine-specific behavior. For controls intended to affect all of the configured Sony control ports, place controls in a file named *vtr* in this directory. For controls intended to affect specific configured Sony control ports, place them in files named *vtr_1*, *vtr_2*, and so on.

Supported Controls

The controls supported by the Sony control module are:

- 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

Set these controls during run-time using the (MVCP) *SSET* and *SGET* commands (not *SET* and *GET*, which are for device controls only). For example:

```
SSET vtr_1 vtr.control.output.idle_mode black
```

Controls that apply to both the underlying media device and the Sony control port take the value set for the control port.

See Appendix B in this document or the vcp-recorder-controls(5) man page for more information concerning the use and default values of these controls.

Unsupported Sony Commands

VST does not currently support the entire Sony protocol specification, including:

- 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: VARIABLE PLAY mode, VAR_FIND, and VAR_REW are fully implemented.

Completing the Configuration

To complete the configuration, for each Sony device controlled by VST, you must enter a line in */usr/vtr/config/control-in.conf*. See Chapter 4, "Configuring Video Server Toolkit" for instructions.

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 VST GUI application "mcclips" (see the *Video Server Toolkit Developer's Guide* 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.

Serial Cabling

For more information about cabling a Sony protocol machine with a Silicon Graphics machine, see Appendix D, "RS-422 Pinouts."

Configuring Your Edit Controller

The delay time for your edit controller must match the delay time set by VST controls. For more information, see "Configuring VST for Your Edit Controller" on page 63.

Opening a Unit

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

Diaquest

Video Server Toolkit (VST) can be used with Diaquest software to control an external deck device. The Diaquest deck control API communicates with an external VTR over a direct RS422 serial link. This API can be used by VST to remotely control an external VTR for frame-accurate capture and lay-down of video-clips.

Using VST and the Diaquest software to control an external VTR requires installation of the *vcp_recorder_eoe.sw.Diaquest* subsystem (see `refer to inst process'). This subsystem contains all the software necessary for the purpose, including the Diaquest supplied software.

Note: There could be licensing requirements to use the Diaquest supplied API library.

The Diaquest software runs on the same server machine as VST and the Diaquest software API is directly invoked by the VST software. Thus a serial port on the VST server is connected to the external VTR using a RS422 link (refer to `cables appendix'). Multiple external decks can be controlled by using a dedicated serial link for each such deck.

The Diaquest API communicates with the external VTR over this link using the sony protocol. Consequently, a single configuration line is required for each such link in VST's control-out.conf file (refer to *control-out.conf*). This line is of the form:

```
sony  rs422  <serial port #>  <baud rate>  <parity>
```

An example entry could be:

```
sony  rs422  3  38400  1
```

Make sure that the external VTR being controlled by VST is put in the slave/remote mode. VST software provides a media port for each such external VTR configured to be controlled by it.

The controls supported by the Diaquest module/port are:

- vtr.deck.error
- vtr.deck.status
- vtr.edit.status
- vtr.edit.preroll
- vtr.edit.postroll
- vtr.media.video.frame_rate
- vtr.media.output.mode
- vtr.edit.coincidence.preroll
- vtr.media.video.sync_port

Refer to Appendix B, "Video Server Toolkit Controls" for a description of these controls.

The *Video Server Toolkit Developer's Guide* describes the MVCP commands used to perform frame-accurate capture of a clip from an external deck onto a VST system.

Installing the Louth Automation Controller

The Louth Video Disk Communications Protocol, defined by Louth Automation, provides full-featured control of the Video Server Toolkit (VST) using RS-232, RS-422, and TCP/IP. The VST Louth processor supports:

- Back-to-back play and record (subject to restrictions imposed by the video I/O port capabilities)

- Archival management

- Control of multiple video (signal) ports from a single communications (control) port

Also consult Chapter 4, "Configuring Video Server Toolkit," for more information about installing Louth in the file *controlin.conf.*

Louth Video Disk Communications Protocol

At the time of this writing, all Louth Video Disk Communications Protocol commands defined as of the February 22, 1995 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 VST

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

Origin/DIVO, Origin/DIVO_DVCPRO, and *O2/O2Video* support simultaneous playing and recording. *Vela/DVB/ATM* supports playback only.

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

- Insert a 422 / 232 protocol converter between the Louth and the O2 workstation.

- 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 and Onyx2 serial ports can run in either RS-232 or RS-422 mode. The configuration file (*/usr/vtr/config/control-in.conf*) specifies which mode to use.

The RS-422 ports are non-standard. See the serial(7) man page for information about the 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 81

2. Editing the *control-in.conf* file.

Configure the Louth ADC-100

On the Louth ADC-100, configure the VST system as a standard video disk. Table 9-1 shows how to specify the device parameters for the video disk.

Table 9-1   Louth Device Parameters

Device Parameter	Setting
AUDIO/VIDEO SWITCHING DATA	All zero
DEVICE NAME	User-specified, for example, louth_0
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
REPORT ITEMS AUTOMATICALLY DELETED OR RECORDED	As desired
DELETE ITEMS AUTOMATICALLY RECORDED FROM DISK AFTER PLAY OUT	Usually disabled

Table 9-1 (continued)   Louth Device Parameters

Device Parameter	Setting
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 the port number in the Louth line in the configuration file, */usr/vtr/config/control-in.conf*.

Completing the Configuration

To complete the configuration, for each Louth device controlled by VST, you must enter a line in */usr/vtr/config/control-in.conf*. See "Serial Port Configuration Lines" on page 48 for instructions, for example, for a RS422 connection:

```
louth rs422 5 38400 1 1 29.97 3
signal 1 DIVO_DVC_0
```

Opening a Unit

Opening a unit owned by a Louth port using MVCP (for example, UOPN) causes unpredictable behavior and is unsupported

Configuring and Using 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. Video Server Toolkit (VST) 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.

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 VST via a serial data connection.

Completing the Configuration

To complete the configuration, for each V-LAN device controlled by VST, you must enter a line in */usr/vtr/config/control-out.conf*, for example:

```
vlan rs422 5 38400
```

See "control-out.conf File" on page 55 for more information.

Installing the Vela Research MPEG-2 Decoder

For MPEG2 decoding, Video Server Toolkit (VST) supports the Vela Research 4-port SCSI-attached decoders. These decoders communicate with the VST server over a Single- or Differential-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 85
- "Connecting the Vela Decoder" on page 86
- "SCSI Ports" on page 86
- "Configuring the Vela Decoder" on page 86

Hardware Requirements

The Vela decoder supports two kinds of SCSI ports: differential and single-ended. Make sure to use the Vela board that supports your 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.

The M-SCSI XIO card has four SCSI ports: port 4 is auto-sensing for single-ended or differential.

VST supports up to three four-port decoders (12 MPEG-2 streams) 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 VST system using a standard SCSI-3 cable. Because 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 need to add an Ultra SCSI PCI adapter to the Origin200 that has an XBox with a four-port Ultra SCSI.

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.

Note: Confirm that the Vela firmware is version 4.35 and later.

When the system is rebooted after connecting the Vela decoders (make sure they are powered on before rebooting the VST 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 ID3 on SCSI bus 1, use this command:

```
% scsicontrol -i sc1d3l0
```

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

When VST 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.

The Vela decoder connected to the first SCSI controller will be *vela_0*, *vela_1*, *vela_2*, and *vela_3*. The Vela decoder connected to the next SCSI controller will be *vela_4*, *vela_5*, *vela_6*, and *vela_7*, and so on.

If you added SCSI controller or moved the SCSIO controller around, the logical part name might change.

To fix the mapping, create the file */usr/vtr/config/vtr_ioconfig.conf*, which fixes the order of the mapping of the Vela decoder connecting to the SCSI controller. The file *vtr_ioconfig.conf* because it has the same function */etc/ioconfig.conf* has for IRIX: freedom to map I/O devices to chosen numbers, for example:

```
1 /hw/module/1/slot/MotherBoard/node/xtalk/8/pci/7/
    scsi.ctlr/0/target/5/lun/0/scsi
```

The first field, 1, is the logical port number. The second field is the hardware graph entry of that part. To generate this file with the default mapping, run */usr/vtr/bin/vtrvclahinv*:

```
% /usr/vtr/bin/vtrvclahinv > /usr/vtr/config/vtr_ioconfig.conf
```

Once this file is generated, adding new hardware does not change the mapping.

If `vtr_ioconfig.conf` exists, the next highest number (2, in the above example) in this file is used for hardware not described in this file.

Instead of using these physical port names, you can provide logical names in the */usr/vtr/config/ports.conf* file, for example:

```
port vela_0 ChA "On-Air Channel A"
port vela_1 ChB "On-Air Channel B"
```

Enabling Genlock

You can use the following control to enable or disable genlock:

```
vtr.media.video.output.sync_source
```

The possible values for the control are *internal*, the default, and *external*. The control may be set using the MVCP SET command or the device-defaults file.

Vela Controls

The following controls and their default values relate to the Vela decoder:

- vtr.media.vela.error.recover_realtime "true"
- vtr.media.vela.error.retry "true"
- vtr.media.vela.output.write_size "65536"
- vtr.media.vela.scsi.timeout "0"
- vtr.media.vela.command_latency "0"
- vtr.media.vela.query_decoder_location "true"
- vtr.media.vela.seamless_playback "false"

For more information about any of these controls, see Appendix B.

Incompatibilities

The vela decoder does not support the following MVCP commands:

- GOTO
- JOG (in reverse)
- SHTL (in reverse)

    Note: SHTL does not provide continuously-variable speeds)

Installing the DVB-ASI Adapter Board

The Viewgraphics MediaPump is a PCI-based, DVB-compliant adaptor board. It multiplexes MPEG 2 transport streams and transmits them over coaxial cables using the Digital Video Broadcast Asynchronous Serial Interface (DVB-ASI) protocol.

This chapter describes how to install and configure a DVB-ASI adapter board for use with Video Server Toolkit (VST).

Hardware Installation

The DVB-ASI adapter board requires a full-length, PCI slot. These slots are supported on the Origin200 but not the O2.

Note: The Origin2000 does not currently support the DVB-ASI adapter board.

Slots and Port Numbers

The */etc/ioconfig.conf* file is automatically generated and maintained. The file shows how PCI card slots correspond to DVB-ASI boards in the Origin200 and Origin2000:

```
0 /hw/module/1/slot/io7/xbox_dualxtown/pci/3/Dynamo
1 /hw/module/1/slot/io1/MotherBoard/pci/5/Dynamo
```

The first line says that DVB-ASI 0 is in PCI slot 3 of the GIGAchannel box. The second line says that DVB-ASI 1is in PCI slot 5 of Module #1 Motherhoard.

Multiple PCI modules (XBOX) can be ganged together.

For more information about *ioconfig.conf*, see "Automatic DVB-ASI Configuration" on page 62.

Software Installation

The software installation for the DVB-ASI board appears as an option in the inst package that comes with VST. To use the DVB-ASI adapter card, you must install *vcp_recorder_eoe.sw.dvb-asi.*

You must also install the Viewgraphics software, *mediapump.sw.* Reboot the machine after this installation.

For more information about installing this software, follow the installation instructions for VST presented in Chapter 3, "Installing Video Server Toolkit."

Configuring DVB-ASI

The DVB-ASI board might work without making any change to its default configuration. If you have resource conflicts, however, the following sections explain which controls to change.

Maximizing Number of Output Streams

To maximize the number of output streams, set the following control:

```
vtr.media.dvb-asi.seamless_playback false
```

The default value is false.

Changing the Video Program ID

You might want to change the video program ID for one of the following reasons:

•   The stream is encoded with one program ID, and you want to send it out with a different program ID.

•   Multiple input streams are encoded with the same program ID, and you want to send them out with different program IDs.

To change the video program ID of a clip, specify the new program ID in the argument of the *vtr.media.mpeg.pid.video* control.

You must change the program ID before cueing a clip. The new program ID remains in effect for that unit even after the clip has finished.

The default value for the control, -1, means that the program ID is unchanged.

Changing the Audio Program ID

You might want to change the audio program ID for one of the following reasons:

* The stream is encoded with one program ID and you want to send it out with a different program ID.
* Multiple input streams are encoded with the same program ID, and you wanted to send them out with different program IDs.

The audio PID is an offset of the video PID.

To change the audio program ID of a clip, specify the new program ID in the argument of the *vtr.media.mpeg.pid_offset.audio* control. The default offset is one.

You must change the program ID before cueing a clip. The new program ID remains in effect for that unit even after the clip has finished.

Changing the PCR Program ID

The Program Clock Reference (PCR) is usually the same as the video program ID. Generally, the PCR should not be changed. You can change the PCR, for example, to carry a separate program ID.

The PCR program ID is an offset of the video PID.

To change the PCR video program ID of a clip, specify the new program ID in the argument of the *vtr.media.mpeg.pid_offset.pcr* control. The default offset is zero.

You must change the PCR program ID before cueing a clip. The new PCR program ID remains in effect for that unit even after the clip has finished.

Specifying the DVB-ASI Board to Use

When you have more than one DVB-ASI board in your system, you can specify the one to use with the *vtr.media.dvb-vg.port* control. In the argument of the control, specify the serial port of the DVB-ASI board you want to use. The default is port zero.

You must set this value before you cue a clip. The specified board remains selected even after the clip has finished.

Incompatibilities

The DVB-ASI adapter board only supports the following mvcp commands:

- CUE
- LOAD
- PLAY
- STOP
- UNLD

Managing Video Server Toolkit

This part of the book describes how to manage Video Server Toolkit (VST). Management tasks include starting and stopping VST, reviewing log messages, installing media clips, and monitoring system performance.

Chapter 13, "Starting and Stopping VST," explains how to start VST.

Chapter 14, "Logging," describes the diagnostic log messages output by VST.

Chapter 15, "Monitoring the System," describes how to monitor VST.

Chapter 16, "Troubleshooting," lists an assortment of common problems and their solutions.

Appendix A, "Example Filesystem and Plexing Setup," describes how to set up a highly available file system.

Appendix B, "Video Server Toolkit Controls," describes the variables, called controls, that help configure the system.

Appendix C, "Automating Video Server Toolkit Commands Using Perl," discusses the */usr/vtr/lib/perl/MVCP.pm* Perl module, which automates back-to-back recording.

Appendix D, "RS-422 Pinouts," describes the pinouts for the RS-422 Mini-DIN8 to DB9 adapters for Silicon Graphics and Apple Macintosh cables.

Starting and Stopping VST

Video Server Toolkit (VST) is normally started automatically when the VST server boots. In a development environment, you may choose to disable automatic startup and, instead, start it manually.

This chapter explains how to start VST in the following sections:

- "Starting VST Manually" on page 95
- "Starting VST at Boot Time" on page 96
- "Starting and Stopping VST" on page 96
- "Confirming the Start of VST" on page 97

To configure start up defaults, see "Start Up Options" on page 60.

Starting VST Manually

If VST does not start automatically, you can start it manually using the following command:

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

vtrstart starts *vtrd* which in turn starts *vvtr*, the VST server software.

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

You should be superuser when you start *vvtr* to ensure reliable operation. If you are not superuser, VST starts but writes a warning message to *vtrlog*. For more information about *vtrlog*, see Chapter 14, "Logging."

Note: When VST 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 workstation that allows you to see the outgoing video in the VST X Window, on the graphics display. This window is sized to display a 625-line video frame with non-square pixels (720 x 576). If you are playing video with a smaller image size, the image is centered within the VST X window.

Note: If you are playing video with non-square pixels (CCIR601 timing), for example an NTSC 525 line video frame, the aspect ratio in the graphics window is incorrect; pixels are square on computer graphics.

The graphics window is enabled by default on O2 workstation. 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 VST automatically at boot time. The *vvtr* X Window appears only when you use */usr/vtr/bin/vtrstart* to start the VST.

To disable automatic startup, enter this command;

```
# chkconfig vtr off
```

Starting VST at Boot Time

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

```
# chkconfig vtr on
```

Starting and Stopping VST

The commands to start and stop VST 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 VST, respectively, from the command line, use the following commands:

```
# vtrstart
# vtrstop
```

Confirming the Start of VST

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

```
# vtrstat
Video Server Toolkit on <host> is running
#
```

vtrstat returns either `running` or `stopped`.

What Happens When VST Crashes

Inside */var/adm/vtr/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 VST detects a program crash, *vvtr* 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.

Note: mcstat and mcclips dump core locally.

Logging

Log messages advise you of the status of Video Server Toolkit (VST). The preset logging messages are associated with levels of warning, from informational messages to messages critical to the running of VST. 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 99
- "Logging Options" on page 100
- "Managing Log Rollover" on page 103
- "VST Log Messages" on page 104

Configuring Logging

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

To modify the VST 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 "Start Up Options" on page 60.

Logging Options

Whether entered at the command line, or as a line of text in */usr/vtr/config/ vtrd.conf*, the VST daemon command takes the logging options described in Table 14-1.

Table 14-1   Logging Options

Flag	Variable	Definition
-f	log-level	Set the maximum log message priority for the log file, */var/adm/vtr/logs/logfile*, 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 102.
-l	log-options	Omit specified log message fields from log messages. For more information, see "Log-Level Options" on page 102.
-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 101.
-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	Sets the severity of messages written to stdout from the process that starts up VST. 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 VST server, */usr/vtr/bin/vvtr*, is started alone. When the server is started by the VST daemon, as it is when the *vtrstart* command is used, all stdout log messages are discarded.

Tip: Setting the logging options to *-f 2 -v 3* provides detailed debugging information in the window in which VST was started and a reasonable amount of debug information in the logfile for after-the-fact analysis. Level 3 fills up the filesystem if applied for a long time to the logfile, but scrolls up and away from the shell window, where you can still see what happened just before a crash in level 3 detail.

Log Levels

-**s** sets the severity level of the messages written to the server log, *syslog*. Table 14-2 shows the VST log severity levels and codes, which are listed in decreasing order of severity.

Table 14-2   Log Levels

Severity Level	Severity Code	Description
Emergency	P	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 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 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 defaults 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 *vvtr* manually. In that case, the log messages go to the window in which VST was invoked.

If VST is invoked automatically at boot time, you should configure the logging level for the server log file (-**f** option) as desired to help troubleshoot VST 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 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 AM if the log file is at least 1 MB large. The default number of log files retained on the system is ten.

The command *rotatelogs* is used to manage the log files and is executed by the system's cron job handler at the right time. Table 14-3 shows the list of available options.

Table 14-3   rotatelogs Command Options

Option	Specifies	Default Value
-b	Log filename	vtrlog
-d	Log file directory	/var/adm/vtr/logs
-h	Lists these options	
-l	Daemon to notify	vtrd
-m	Maximum number of backups retained on the machine	10
-s	Minimum log file size to trigger rollover	1 MB
-D	Debug level	Off

Changing Default Values

To change any defaults (such as the time of the rollover or the log size), the crontab entry of rotatelogs in the system must be edited (as root, using *crontab -e*). This entry was installed as part of the initial VST installation by an exitop.

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

```
1       2       *       *       *       /usr/vtr/bin/rotatelogs -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).

VST Log Messages

Each VST 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.

Monitoring the System

Once Video Server Toolkit (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. You can also use standard Silicon Graphics IRIX tools to monitor various system resources.

There are two tools that monitor VST directly:

- vtrstat
- mcstat

mcstat is the graphical version of *vtrstat*. Both tools monitor VST operations.

This chapter describes how to monitor VST in the following sections:

- "vtrstat" on page 105
- "mcstat" on page 107
- "IRIX Tools" on page 107

vtrstat

vtrstat is a command-line tool that tells you:

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

Example 15-1 shows an example output of *vtrstat*.

Example 15-1   vtrstat Output

```
kruger 87# /usr/vtr/bin/vtrstat -units
Video Server Toolkit 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
Video Server Toolkit 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
```

A description of these functions can be found in the *Video Server Toolkit Developer's Guide* in Chapter 8, "Completing Common Tasks Using MVCP Commands."

mcstat

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

- Whether or not VST is running

- Which units are open

- Which media ports are available

Figure 15-1 shows an example of *mcstat*'s output when VST is running. When VST is not running, mcstat exits.

Figure 15-1   mcstat Output

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

IRIX Tools

Silicon Graphics provides a collection of monitoring tools that can be used with VST, including

- sar—system activity reporter. Reports operating system activity.

- gr_osview—graphical system monitor. Graphically displays real-time usage of certain system resources.

- osview—a text version of gr_osview.

- 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 VST to IRIX tools, you must manipulate the data. For more information about exporting media data, see *Video Server Toolkit Developer's Guide.*

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 15-2 shows an example output of *sar*.

Example 15-2   Output of SAR Command

```
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 use. This display provides a real-time window into the overall operation of the system, as shown in Figure 15-2.

Figure 15-2   gr_osview

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

Using osview

osview is a text version of *gr_osview*. The following show example output of *osview:*

```
Osview 2.1 : One Second Average   vcpqa2      10/02/98 09:30:25 #14
int=5s
 Load Average            4MB pages   0    vidintr       0
   1 Min       0.008     16MB pages  0    drop_add      0
   5 Min       0.000  System Activity    TCP
   15 Min      0.000     syscall   206    conns         0
 CPU 0 Usage             read        3    sndtotal      0
   %user       0.00      write       0    rcvtotal      0
   %sys        1.20      fork        0    sndbyte      38
   %intr       0.00      exec        0    rcvbyte       0
   %sxbrk      0.00      readch     45  UDP
   %idle      98.80      writech    78    ipackets      1
 CPU 1 Usage             iget        0    opackets      0
   %user       0.00   Block Devices       dropped       0
   %sys        0.60      lread       0    errors        0
   %intr       0.00      bread       0  IP
   %sxbrk      0.00      %rcache   0.0    ipackets      1
   %idle      99.40      lwrite      0    opackets      0
 Wait Ratio              bwrite  22.4K    forward       0
   %IO         0.0       wcancel     0    dropped       0
   %Swap       0.0       %wcache   0.0    errors        0
   %Physio     0.0       phread      0  NetIF[ef0]
 System Memory           phwrite     0    Ipackets      2
   Phys      512.0M *Swap                 Opackets      0
    kernel    26.7M *System VM            Ierrors       0
     heap      7.7M  Memory Faults        Oerrors       0
      stream  48.0K     vfault      0     collisions    0
     zones     7.7M     protection  0  *NetIF[lo0]
     ptbl      2.9M     demand      0   Scheduler
     fs ctl    9.8M     cw          0     runq          0
     fs data  105.9M    steal       0     swapq         0
     delwri       0     onswap      0     switch       65
     free     350.5M    oncache     0  *Large page stats
     userdata  19.2M    onfile      0  *Interrupts
     pgallocs     0     freed       0  *PathName Cache
 Node[0]                unmodswap   0  *Heap
     Totalmem505.1M     unmodfile   0  *EfsAct
     freemem  350.5M    iclean      0  *XfsAct
     64k page   1.6K *TLB Actions      *Getblk
     256k pages    0  Video             *Vnodes
exit 1MB pages    0     vidioctl    0
```

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 15-3.
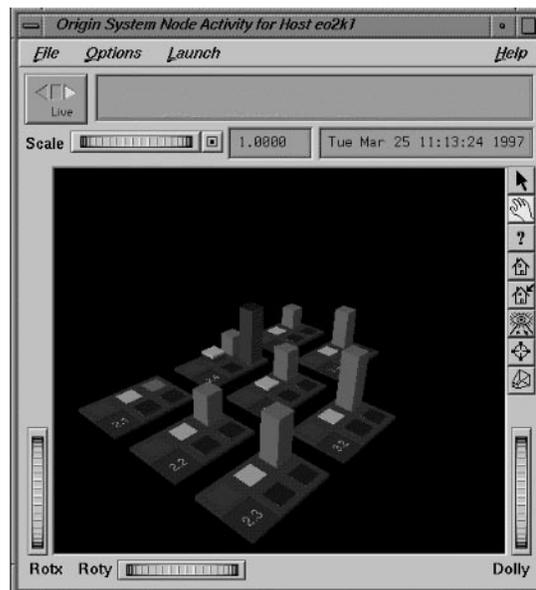
Figure 15-3   Performance CoPilot

For more information about Performance Co-Pilot, see its man page.

Troubleshooting

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

The problems solved include:

- "Exceeding Semaphore Limit" on page 113
- "625/50 Clips Do Not Play" on page 114
- "DIVO Left in Bad State" on page 114
- "Error When Installing VCP-R Version 1.1 Over Version 1.0 or Beta" on page 114
- "Audio Sync Problems When Audio is Missing" on page 115
- "VVTR Crashes Without Audio" on page 115
- "Why Does It Take So Long to Add a Clip?" on page 115

Exceeding Semaphore Limit

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:

1. Edit */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.

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 */usr/vtr/config/system-defaults/main*:

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

This control sets the timing of the system, including the DIVO cards, to 625/50.

Make sure, in */usr/vtr/config/device-defaults/DIVO*, the control, *vtr.media.video.output.timing*, is set to its default value, *system*.

DIVO Left in Bad State

If you find that the DIVO board has been left in a bad state, for example, you cannot reset it, use the *divo_reset* command, included in the *divo.sw.diag* images, to correct the problem. For example, the DIVO reset command is in the directory, */usr/dmedia/DIVO/bin*.

If you are using a 625/50 system, see "625/50 Clips Do Not Play" on page 114.

Error When Installing VCP-R Version 1.1 Over Version 1.0 or Beta

Any attempt to start VST fails.

To solve the problem:

1. Remove VST and the DIVO images.

2. Re-install all the latest required IRIX patches.

3. Re-install the VST images plus the patches that are included on the CDROM.

4. Perform an autoconfig.

5. Reboot the server.

Audio Sync Problems When Audio is Missing

vvtr crashes after a few seconds if you do not have an audio source.

if you are having audio sync problems when there is no audio, do the following:

1. Turn the audio off: in the */usr/vtr/config/device-defaults* directory, edit the DIVO file and add:

    ```
    vtr.media.audio.input.channels.clip "0"
    ```

2. If you do not have an audio source, disable input audio by setting the unit control:

    ```
    vtr.media.audio.input.port ""
    ```

VVTR Crashes Without Audio

3. vvtr crashes after a few seconds if you do not have an audio source. To fix this problem, disable input audio by setting the unit control:

    ```
    vtr.media.audio.input.port ""
    ```

Why Does It Take So Long to Add a Clip?

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

Example Filesystem and Plexing Setup

This appendix describes how to set up a highly available file system for a Video Server Toolkit (VST) system using the following steps:

Sample Hardware Configuration

The installation described in this appendix is based on the following hardware:

- Origin200 with GIGAchannel and 2 modules

- Louth controller

- 4 DIVO_DVC cards

- 4 RAD cards (for audio)

- 2 Ciprico 7000 RAIDS

- 2 PCI FibreChannel cards

- M-serial card

- XIO with 5 serial ports

Overview of the Repartitioning Procedure

In the following installation, you complete the following steps:

1. Load the miniroot from the IRIX 6.4 patch CD.

2. mkfs /dev/dsk/dks0d1s0.

3. Install IRIX 6.4 from the OS CD.

4. Install IRIX 6.4 Applications from the Apps CD.

5. Install IRIX 6.4 patches.

6. Restart the system.

7. Install further applications, patches, and non-Silicon Graphics software from multiuser mode.

Repartitioning the System Disk

Use this section to repartition the disk and to install all software required for the installation of VST.

To do the repartitioning and software installing, you can use either a remote or local CDROM. If you are using a local CDROM drive, skip to "Using a Local CDROM-- Start Here" on page 120.

Using a Remote CDROM--Start Here

If you are using a remote CDROM drive, start here.

1. Set the netaddr NVRAM variable to the IP address in the same subnet range on the system where the CDROM is mounted.

   ```
   # setenv netaddr IPaddress
   ```

   IPaddress is any unique IP address.

2. Enter **exit**.

3. Edit the */etc/inetd.conf* file to allow the remote system to boot from the CDROM directory by appending /CDROM to the configuration line, as shown:

```
tftp    dgram   udp     wait    guest   /usr/etc/tftpd  tftpd -s
/usr/local/boot /CDROM
```

4. Make sure the (11/98 or later) patch set CD for IRIX 6.4 is loaded.

   If it is not loaded, load it now. The miniroot on the IRIX 6.4 CD cannot be used. This patch must be loaded before the machine is rebooted.

5. Restart inetd on the remote system:

   ```
   /etc/killall -HUP inetd
   ```

   The system displays the System Maintenance Menu from the local host.

   ```
   System Maintenance Menu

   1) Start System
   2) Install System Software
   3) Run Diagnostics
   4) Recover System
   5) Enter Command Monitor
   ```

6. Enter 5, the Enter Command Monitor option (option 5).

7. Boot fx:

   ```
   >> boot -f bootp()remotehost:/CDROM/stand/fx.64 --x
   ```

   remotehost is the name of the remote host.

   The system responds:

   ```
   Obtaining /CDROM/stand/fx.64 from server remoteHost
   768+121980+27669+10928 entry: 0xa8000000010455e0
   SGI Version 6.4 ARCS   Jan 23, 1997
   fx: "device-name" = (dksc)
   ```

8. Skip to step 5 on page 120, "Press the Enter key."

Using a Local CDROM-- Start Here

If you are using a local CDROM drive, start here.

1. Make sure the patch set CD for IRIX 6.4 is loaded to load the miniroot.

2. Restart the system using the following command:

   ```
   # init 0
   ```

   The system displays the System Maintenance Menu from the local host.

   ```
   System Maintenance Menu

   1) Start System
   2) Install System Software
   3) Run Diagnostics
   4) Recover System
   5) Enter Command Monitor
   ```

3. Enter **5**, the Enter Command Monitor option (option 5).

4. Boot fx:

   ```
   >> boot -f bootp()/CDROM(0, 4, 7)/stand/fx.64 --x
   ```

   remotehost is the name of the remote host.

   The system responds:

   ```
   Obtaining /CDROM/stand/fx.64 from server remoteHost
   768+121980+27669+10928 entry: 0xa8000000010455e0
   SGI Version 6.4 ARCS   Jan 23, 1997
   fx: "device-name" = (dksc)
   ```

5. Press the Enter key.

   The system responds:

   ```
   fx: ctlr# = (0)
   ```

6. Press the Enter key.

   The system responds:

   ```
   fx: drive# = (1)
   ```

7. Press the Enter key.

    The system responds

    ```
    ...opening dksc(0,1,0)
    ...drive selftest...OK
    Scsi drive type == SGI     QUANTUM XP39100WLXY7

    ----- please choose one (? for help, .. to quit this menu)-----
    [exi]t             [d]ebug/            [l]abel/            [a]uto
    [b]adblock/        [exe]rcise/         [r]epartition/      [f]ormat
    fx>
    ```

8. Enter **r** for repartition.

    The system responds:

    ```
    ----- partitions-----
    part  type          blocks              Megabytes   (base+size)
      0: xfs          4096 + 17253136         2 + 8424
      1: raw     17257232 + 524288        8426 + 256
      8: volhdr         0 + 4096             0 + 2
     10: volume         0 + 17781520         0 + 8682

    capacity is 17781520 blocks

    ----- please choose one (? for help, .. to quit this menu)-----
    [ro]otdrive            [o]ptiondrive          [e]xpert
    [u]srrootdrive         [re]size
    fx/repartition>
    ```

9. Enter **e -b**.

    The system responds:

    ```
    Warning: you will need to re-install all software and restore user
    data from backups after changing the partition layout.  Changing
    partitions will cause all data on the drive to be lost.  Be sure
    you have the drive backed up if it contains any user data.
    Continue?
    ```

10. Enter **yes**.

    The system responds:

    ```
    Enter .. when done
    fx/repartition/expert: change partition = (0)
    ```

11. Press the Enter key.

    The system responds:

    ```
    before:  type xfs         block     4096,          2 MB
                              len:  17515280 blks, 8552 MB
    fx/repartition/expert: partition type = (xfs)
    ```

12. Enter **xlv**

    The system responds:

    ```
    fx/repartition/expert: first block = (4096)
    ```

13. Press the Enter key.

    The system responds:

    ```
    fx/repartition/expert: number of blocks (max 17777424) = (17515280)
    ```

14. Enter **524096**

    The system responds:

    ```
    after:  type xlv          block     4096,          2 MB
                              len:     524096 blks,  256 MB
    fx/repartition/expert: change partition = (1)
    ```

15. Press the Enter key.

    The system responds:

    ```
    before:  type raw         block 17519376,       8554 MB
                              len:     262144 blks,  128 MB
    fx/repartition/expert: partition type = (raw)
    ```

16. Press the Enter key.

    The system responds:

    ```
    fx/repartition/expert: first block = (17519376)
    ```

17. Enter **528192**

    The system responds:

    ```
    fx/repartition/expert: number of blocks (max 17253328) = (262144)
    ```

18. Enter **1048192**

    The system responds:

    ```
    after:  type raw        block  528192,       258 MB
                             len:  1048192 blks,  512 MB
    fx/repartition/expert: change partition = (6)
    ```

19. Press the Enter key.

    The system responds:

    ```
    before:  type xfs        block       0,        0 MB
                             len:        0 blks,    0 MB
    fx/repartition/expert: partition type = (xfs)
    ```

20. Enter **xlv**.

    The system responds:

    ```
    fx/repartition/expert: first block = (0)
    ```

21. Enter **1576384**

    The system responds:

    ```
    fx/repartition/expert: number of blocks (max 16205136) = (0)
    ```

22. Enter **14107984**

    The system responds:

    ```
    after:  type xlv        block 1576384,       770 MB
                             len:  14107984 blks, 6889 MB
    fx/repartition/expert: change partition = (7)
    ```

23. Press the Enter key.

    The system responds:

    ```
    before:  type xfs        block       0,        0 MB
                             len:        0 blks,    0 MB
    fx/repartition/expert: partition type = (xfs)
    ```

123

24. Enter **xlv**.

    The system responds:

    ```
    fx/repartition/expert: first block = (0)
    ```

25. Enter **15684368**
    The system responds:

    ```
    fx/repartition/expert: number of blocks (max 2097152) = (0)
    ```

26. Enter **2097152**

    The system responds:
    ```
     after:  type xlv         block 15684368,      7658 MB
                              len:  2097152 blks, 1024 MB
    fx/repartition/expert: change partition = (8)
    ```

27. Enter two periods (..).

    The system responds:

    ```
    ----- partitions-----
    part  type          blocks              Megabytes   (base+size)
      0: xlv         4096 + 524096            2 + 256
      1: raw       528192 + 1048192         258 + 512
      6: xlv     1576384 + 14107984         770 + 6889
      7: xlv    15684368 + 2097152         7658 + 1024
      8: volhdr        0 + 4096              0 + 2
     10: volume        0 + 17781520          0 + 8682

    capacity is 17781520 blocks

    ----- please choose one (? for help, .. to quit this menu)-----
    [ro]otdrive           [o]ptiondrive           [e]xpert
    [u]srrootdrive        [re]size
    fx/repartition>
    ```

28. Enter **/l/sync**

    The system responds with the following information for partitioning two 9 GB disks:

    ```
    writing label info to dksc(0,1,0)

    ----- partitions-----
    part  type          blocks             Megabytes   (base+size)
      0: xlv        4096 + 524096          2 + 256
      1: raw      528192 + 1048192       258 + 512
      6: xlv     1576384 + 14107984      770 + 6889
      7: xlv    15684368 + 2097152      7658 + 1024
      8: volhdr        0 + 4096            0 + 2
     10: volume        0 + 17781520        0 + 8682

    capacity is 17781520 blocks

    ----- please choose one (? for help, .. to quit this menu)-----
    [ro]otdrive          [o]ptiondrive           [e]xpert
    [u]srrootdrive       [re]size
    fx/repartition>
    ```

29. Enter **/exit.**

    The system displays the System Maintenance menu.

    ```
    System Maintenance Menu

    1) Start System
    2) Install System Software
    3) Run Diagnostics
    4) Recover System
    5) Enter Command Monitor
    ```

30. Enter **2**.

    The system responds:

    ```
                    Installing System Software...

            Press <Esc> to return to the menu.

    1) Remote Directory  X) Local CD-ROM

    Enter 1-2 to select source type, <esc> to quit,
    or <enter> to start:
    ```

31. If you are installing from a remote directory, enter **1**. If you are installing from a local CDROM, enter **2**. and skip to step 34, "Press the Enter key."

    The system responds:

    ```
    Enter the name of the remote host:
    ```

32. Enter the name of the remote host.

    The system responds:

    ```
    Enter the remote directory:
    ```

33. Enter **/CDROM/dist**.

    The system responds:

    ```
    1)[Remote Directory]  X) Local CD-ROM
          *a) Remote directory /CDROM/dist from server remoteHost.

    Enter 1-2 to select source type, a to select the source, <esc> to
    quit, or <enter> to start:
    ```

34. Press the Enter key.

    The system responds:

    ```
    Copying installation program to disk.
    ......... 10% ......... 20% ......... 30% ......... 40% ....... 50%
    ......... 60% ......... 70% ......... 80% ......... 90% ....... 100%

    Copy complete
    IRIX Release 6.4 IP27 Version 02121744 System V - 64 Bit
    Copyright 1987-1997 Silicon Graphics, Inc.
    All Rights Reserved.

    root on
    /hw/module/1/slot/MotherBoard/node/xtalk/8/pci/0/scsi_ctlr/0/target/
    1/lun/0/disk/partition/1/block ; boot swap file on /dev/swap swplo
    55000
    WARNING: Cannot load runtime symbol table from bootp'ed kernel.
             Loadable modules will not be registered or loaded.
    Creating miniroot devices, please wait...

    Current system date is Thu Dec 10 13:51:28 PST 1998

    Mounting file systems:

    mount: /dev/dsk/dks0d1s0 on /root: Filesystem too large for device.
    mount: giving up on:
        /root

    Unable to mount partition: /dev/dsk/dks0d1s0 on /root.
    This is your system disk: without it we have nothing
    on which to install software.

    Please manually correct your configuration and try again.

            Press Enter to invoke C Shell csh:
    ```

35. Press the Enter key.

    The system responds:

    ```
    #
    ```

127

36. Make a filesystem for the */root* partition. Enter *mkfs /dev/dsk/dks0d1s0*

    The system responds:

    ```
    meta-data=/dev/dsk/dks0d1s0        isize=256    agcount=8,
    agsize=8189 blks
    data      =                        bsize=4096   blocks=65512,
    imaxpct=25
    log       =internal log            bsize=4096   blocks=1000
    realtime =none                     extsz=65536  blocks=0, rtextents=0
    #
    ```

37. Make a filesystem for the */usr* partition. Enter *mkfs* /dev/dsk/dks0d1s6
    The system responds:
    ```
    meta-data=/dev/dsk/dks0d1s6        isize=256    agcount=8,
    agsize=220438 blks
    data      =                        bsize=4096   blocks=1763498,
    imaxpct=25
    log       =internal log            bsize=4096   blocks=1000
    realtime =none                     extsz=65536  blocks=0, rtextents=0
    #
    ```

38. Enter **exit**.

    The system responds:

    ```
    Invoking software installation.

    What is the hostname of this machine?
    ```

39. Enter the name of the host machine.

    The system responds:

    ```
    What is the network address of hostMachine?
    ```

40. Enter the network address of the host.

    The system responds:

    ```
    Starting network with hostname: hostMachine, at ip address:
    198.29.66.72

    Default distribution to install from: CDROMHost:/CDROM/dist

    For help on inst commands, type "help overview".

    Inst Main Menu

     1. from [source ...]          Specify location of software to be
    installed
     2. open [source ...]          Specify additional software
    locations
     3. close [source ...]         Close distributions
     4. list [keywords] [names]    Display information about software
    subsystems
     5. go                         Perform software installation and
    removal now
     6. install [keywords] [names] Select subsystems to be installed
     7. remove [keywords] [names]  Select subsystems to be removed
     8. keep [keywords] [names]    Do not install or remove these
    subsystems
     9. step [keywords] [names]    Interactive mode for
    install/remove/keep
    10. conflicts [choice ...]     List or resolve installation
    conflicts
    11. help [topic]               Get help in general or on a
    specific word
    12. view ...                   Go to the View Commands Menu
    13. admin ...                  Go to the Administrative Commands
    Menu
    14. quit                       Terminate software installation

        Inst>
    ```

41. Load the IRIX 6.4 CD into the CDROM drive.

42. Enter **sh**.

    The system responds:

    ```
    #
    ```

43. Enter **mount /dev/dsk/dks0d1s6 /root/usr.**

129

44. Enter **exit**.

45. Enter 1.

    ```
    Previous installation sites:

     1 CDROMHost:/CDROM/dist
     2 none (no distribution, view installed products)
     3 quit (no action)

    Install software from: [remoteHost:/CDROM/dist]
    ```

46. Press the Enter key.

    The system responds:

    ```
    What is the network address of hostName?
    ```

47. Enter the network address of the host.
    The system responds after help text:
    ```
    Reading product descriptions .. 100% Done.

    Inst>
    ```

48. Enter go.

    The operating system installs and the system responds:

    ```
    inst?
    ```

49. Install the (non-default) images for xfs and xlv:

    ```
    inst> i eoe.sw.xfs* eoe.sw.xlv*
    ```

    The system responds:

    ```
    ...
    Do you wish to run the optional installation startup script?
    1. Run the installation startup script now
    2. Ignore the startup script
    3. Show the contents of the startup script
    4. Help/Show the introductory "README" statement again
    Please enter a choice [1]:
    ```

50. Press the Enter key.

    The system responds:

    ```
    Inst>
    ```

130

51. Place the IRIX 6.4 Recommended/Required Patches CD into the CDROM drive.

    Do not install the IRIX 6.5 patches.

52. Install the patches.

```
Inst> install installable
Inst> keep same
Inst> go
```

⚠️ **Warning:** **The 6.4 patches must be installed before the machine is rebooted.**

53. Enter `quit`.

    The system responds:

```
Requickstarting ELF files (see rqsall(1)) .. 100% Done.
Automatically reconfiguring the operating system.
Ready to restart the system.  Restart? { (y)es, (n)o, (sh)ell,
(h)elp }:
```

54. Reboot the *hostMachine*:

```
# reboot hostMachine
```

    The system responds:

```
Reboot hostMachine?
```

55. Enter `y`.

    The system responds:

```
****    Enter Reason for Shutdown  ****
Please select one of the following choices by number:

1. Environmental reasons (move system, planned power outage, ... )
2. Upgrade/add hardware or peripherals
3. Replace failed hardware component
4. Install software that requires reboot
5. Scheduled reboot/Routine maintenance
6. Software problem
7. Other unclassified reasons

Enter Number (within 30 seconds):
```

56. Enter **4**.

The system restarts.

57. Enter the following commands to make sure that the */var/tmp* directory resides on the */usr* partition:

```
mkdir /usr/vtrtmp
rm -r /var/tmp
rm -r /usr/tmp
ln -s /usr/vtrtmp /var/tmp
ln -s /var/tmp /usr/tmp
```

Install VST Now

Install the VST software now. Refer to Chapter 3, "Installing Video Server Toolkit."

Plexing Setup

Use the following procedure to set up the plexing after installing VST.

1. Make sure that plexing is enabled and supported. Start xlv and run the show *config* command:

```
# xlv_mgr
xlv_mgr> show config
```

The system responds:

```
Plexing license: present
Plexing support: present
```

If a plexing license is not present, exit xlv_mgr to install the license; the license should be installed in */var/flexlm/license.dat* and then provide a symlink to the */etc/flexlm* directory.

```
# ln -s /var/flexlm /etc/flexlm
```

For more information about installing flexlm, see "Flexlm License Installation Instructions" on page 145.

2. Exit xlv_mgr

```
xlv_mgr> exit
```

3. Duplicate the partitioning on the system disk to the option disk.

4. Enter

   ```
   # fx -x
   ```

5. The system responds:

   ```
   fx version 6.4, Sep 17, 1997
   fx: "device-name" = (dksc)
   ```

6. Press the Return key

   The system responds:

   ```
   fx: ctlr# = (0)
   ```

7. Press the Enter key.

   The system responds:

   ```
   fx: drive# = (1)
   ```

8. Enter **2**.

   The system responds

   ```
   ...opening dksc(0,1,0)
   ...drive selftest...OK
   Scsi drive type == SGI     QUANTUM XP39100WLXY7

   ----- please choose one (? for help, .. to quit this menu)-----
   [exi]t              [d]ebug/            [l]abel/            [a]uto
   [b]adblock/         [exe]rcise/         [r]epartition/      [f]ormat
   fx>
   ```

9. Repeat steps 8 on page 121 through and including step 29 on page 125.

10. Make the root partition on the system disk into an XLV volume. Enter:

    ```
    # xlv_make
    ```

    The system responds:

    ```
    xlv_make>
    ```

11. Enter **vol root.**

    The system responds:
    ```
    xlv_make>
    ```

12. Enter **data.**

    ```
    root.data
    xlv_make>
    ```

13. Enter **ve -force /dev/dsk/dks0d1s0.**

    The system responds:

    ```
    root.data.0.0
    xlv_make>
    ```

14. Enter **end**.

    The system responds:

    ```
    Object specification completed
    xlv_make>
    ```

15. Enter vol **usr**.

    The system responds:

    ```
    usr
    xlv_make>
    ```

16. Enter **data**.

    The system responds:

    ```
    usr.data
    xlv_make>
    ```

17. Enter **ve -force /dev/dsk/dks0d1s6**.

    The system responds:

    ```
    usr.data.0.0
    xlv_make>
    ```

18. Enter **end**.

    The system responds:

    ```
    Object specification completed
    xlv_make>
    ```

19. Enter **exit**.

    The system responds:

    ```
    Newly created objects will be written to disk.
    Is this what you want?(yes)
    ```

20. Enter `yes`.

    The system responds:

    ```
    Invoking xlv_assemble
    ```

    The result is XLV volumes named root and usr that contains the root and usr partitions on the system disk. Since XLV preserves the data in partitions, the contents of the root and usr partitions are preserved. The -force option to the ve command was used because a mounted partition was included in the volume.

21. Edit */etc/fstab* and add the following entry:

    ```
    /dev/xlv/usr/    /usr    xfs    rw    0    0
    ```

22. Create a */etc/config/xlv_plexd.options* file with the following content:

    ```
    # cat > /etc/config/xlv_plexd.options
      -b 1024 -w 50

    Press ctrl-D   # to close the file
    ```

    These commands make sure that a revive does not use to much of the disk's bandwidth.

23. Reboot the system so that the system switches from running off the root partition on the system disk i.e. */dev/dsk/dks0d1s0*, to running off the logical volume */dev/xlv/root*:

    ```
    # reboot
    ```

24. Create the plex for the root out of /dev/dsk/dks0d2s0, and call it
    ```
     root_plex; also create plex for usr out of /dev/dsk/dks0d2s6, and
     call it usr_plex
    ```

    ```
    # xlv_make
    ```

    The system responds:

    ```
    xlv_make>
    ```

25. Enter `plex root_plex`.

    The system responds:

    ```
    root_plex
    xlv_make>
    ```

26. Enter `ve /dev/dsk/dks0d2s0`.

    The system responds:

    ```
    root_plex.0
    xlv_make>
    ```

27. Enter **end**.

    The system responds:

    ```
    Object specification completed
    xlv_make>
    ```

28. Enter **plex usr_plex**.

    The system responds:

    ```
    usr_plex
    xlv_make>
    ```

29. Enter **ve /dev/dsk/dks0d2s6**.

    The system responds:

    ```
    usr_plex.0
    xlv_make>
    ```

30. Enter **end**.

    The system responds:

    ```
    Object specification completed
    xlv_make>
    ```

31. Enter **exit**.

    The system responds:

    ```
    Newly created objects will be written to disk.
    Is this what you want?(yes)
    ```

32. Enter **yes**.

    The system responds:

    ```
    Invoking xlv_assemble
    ```

33. Add *sash* to the volume header of the option disk used for the root_plex. It enables booting off of the root_plex if the primary root fails.

    ```
    # dvhtool -v get sash /tmp/sash /dev/rdsk/dks0d1vh
    # dvhtool -v add /tmp/sash sash /dev/rdsk/dks0d2vh
    ```

34. Attach the root_plex to the root volume, and usr_plex to the usr volume using *xlv_mgr*.

    ```
    # xlv_mgr
    ```

    The system responds:

    ```
    xlv_mgr>
    ```

35. Enter the following three lines in xlv_mgr:

    ```
    xlv_mgr> attach plex root_plex root.data
    xlv_mgr> attach plex usr_plex usr.data
    xlv_mgr> quit
    ```

    When the shell prompt returns, the system automatically begins a plex revive so that the two plexes contain the same data.

36. Determine the SCSI ID and bus info of the Ciprico FC disk arrays.

    ```
    # hinv
    ```

    The system responds:

    ```
    FPU: MIPS R10010 Floating Point Chip Revision: 0.0
    CPU: MIPS R10000 Processor Chip Revision: 2.6
    2 180 MHZ IP27 P
    ...

    Integral SCSI controller 3: Version Fibre Channel AIC-1160,
    revision 2
    Disk drive: unit 1 on SCSI controller 3
    Integral SCSI controller 2: Version Fibre Channel AIC-1160,
    revision 2
    Disk drive: unit 1 on SCSI controller 2
    ...

    DIVO Video: controller 2 unit 2: Input, Output
    DIVO Video: controller 3 unit 3: Input, Output
    IOC3 external interrupts: 1
    ```

37. Start fx.

    ```
    # fx -x
    ```

38. Enter partition data:

    ```
    fx: "device-name" = (dksc)
    ```

39. Press the Return key.

    The system responds:

    ```
    fx: ctlr# = (0)
    ```

40. Enter controller where the Ciprico is, for example, **3**.

    The system responds:

    ```
    fx: drive# = (1)
    ```

41. Press the Return key.

    The system responds:

    ```
    fx: lun# = (0)
    ```

42. Press the Return key.

    The system responds:

    ```
    ----- please choose one (? for help, .. to quit this menu)-----
    [exi]t             [d]ebug/           [l]abel/            [a]uto
    [b]adblock/        [exe]rcise/        [r]epartition/      [f]ormat
    fx>
    ----- partitions-----
    part  type          blocks              Megabytes   (base+size)
      7: xlv         4096 + 142258752         2 + 69462
      8: volhdr         0 + 4096             0 + 2
     10: volume         0 + 142262848         0 + 69464

    capacity is 142262848 blocks

    ----- please choose one (? for help, .. to quit this menu)-----
    [ro]otdrive           [o]ptiondrive           [e]xpert
    [u]srrootdrive        [re]size
    fx/repartition>
    ```

43. Enter o, for optiondrive.

    The system responds:

    ```
    fx/repartition/optiondrive: type of data partition = (xfs)
    ```

44. Press the Enter key.

    The system responds:

    ```
    fx/repartition/optiondrive: create log partition? = (no)
    ```

45. Press the Enter key.

    The system responds:

    ```
    Warning: you will need to re-install all software and restore user
    data
    from backups after changing the partition layout.  Changing
    partitions
    will cause all data on the drive to be lost.  Be sure you have the
    drive
    backed up if it contains any user data.  Continue? yes

    ----- partitions-----
    part  type          blocks              Megabytes  (base+size)
      7: xlv         4096 + 142258752         2 + 69462
      8: volhdr         0 + 4096             0 + 2
     10: volume         0 + 142262848        0 + 69464

    capacity is 142262848 blocks

    ----- please choose one (? for help, .. to quit this menu)-----
    [ro]otdrive          [o]ptiondrive          [e]xpert
    [u]srrootdrive       [re]size
    fx/repartition>
    ```

46. Enter **/l/sync.**

    The system responds:

    ```
    writing label info to dksc(0,3,0)

    ----- partitions-----
    part  type          blocks              Megabytes  (base+size)
      7: xlv         4096 + 142258752         2 + 69462
      8: volhdr         0 + 4096             0 + 2
     10: volume         0 + 142262848        0 + 69464

    capacity is 142262848 blocks

    ----- please choose one (? for help, .. to quit this menu)-----
    [ro]otdrive          [o]ptiondrive          [e]xpert
    [u]srrootdrive       [re]size

    fx/repartition>
    ```

47. Enter **/exit**.

48. Repeat steps 38 on page 137 through step 47 on page 139 for each Ciprico RAID.

49. Create the xlv volume for the clip cache using the data partitions on the system and option disk, and the two fibre channel raids. Enter:

```
# xlv_make
```

The system responds:

```
name the volume:
xlv_make>
```

50. Enter the volume's name, for example, enter **vol vtr**.

The system responds:

```
vtr
xlv_make>
```

51. Enter **data**.

The system responds:

```
vtr.data
xlv_make>
```

52. Enter **plex**.

The system responds:

```
vtr.data.0
xlv_make>
```

53. Enter **ve dks0d1s7**.

The system responds:

```
vtr.data.0.0
xlv_make>
```

54. Enter **plex**.

The system responds:

```
vtr.data.1
xlv_make>
```

55. Enter **ve dks0d2s7**.

The system responds:

```
vtr.data.1.0
xlv_make>
```

56. Enter **rt**.

    The system responds:

    ```
    vtr.rt
    xlv_make>
    ```

57. Enter **plex**.

    The system responds:

    ```
    vtr.rt.0
    xlv_make>
    ```

58. Enter **ve -stripe -stripe_unit 4096 /dev/dsk/dks2d1s7 /dev/dsk/dks3d1s7**.

    The system responds:

    ```
    vtr.rt.0.0
    xlv_make>
    ```

59. Enter **end**.

    The system responds:

    ```
    Object specification completed
    xlv_make>
    ```

60. Enter **exit**.

    The system responds:

    ```
    Newly created objects will be written to disk.
    Is this what you want?(yes)
    ```

61. Enter **yes**.

    The system responds:

    ```
    Invoking xlv_assemble
    ```

62. Create an xfs filesystem on the clip's logical volume.

    ```
    # mkfs_xfs -b size=16k -r extsize=4m /dev/xlv/vtr
    ```

63. Create an entry in the filesystem table

    ```
    # vi /etc/fstab
    /dev/xlv/vtr  /usr/vtr/clips xfs     rw,raw=/dev/rxlv/vtr  0 0
    ```

64. Mount the clips filesystem.

    ```
    # mount -av
    ```

65. Create required links for */var/adm* and */var/tmp:*

```
# mkdir /usr/vtrtmp
# rm -r /var/tmp
# rm -r /usr/tmp
# ln -s /usr/vtrtmp /var/tmp
# ln -s /var/tmp /usr/tmp
```

66. Using the SCSI ID and bus information determined in step 36, configure GRIO by running *grio_bandwidth* for 4MB I/O sizes:

```
# grio_bandwidth -d /dev/dsk/dks3d1s7 -t 60 -u -s 4194304
```

The system responds:

```
device
/hw/module/1/slot/io7/MotherBoard/pci/3/scsi_ctlr/0/target/1/lun/0/
    disk with ID "Ciprico Rimfire 7010    03.1",
using only read ops, provided:
on average:          10 ops of size 4194304 each second
in the worst case:    9 ops of size 4194304 each second
```

67. Add this information to */etc/grio_disks*:

```
ADD    "Ciprico Rimfire 7010    03.1"  4096K 11
```

68. Use the */hw* path information obtained from *grio_bandwidth* to put the following in the */etc/grio_disks* file:

```
REPLACE
/hw/module/1/slot/io7/MotherBoard/pci/3/scsi_ctlr/0/target/1/lun/0/d
    isk 4194304 11

REPLACE /hw/module/1/slot/io7/MotherBoard/pci/3/scsi_ctlr/0 4194304
    25
```

69. Repeat steps 66 on page 142 through step 68 on page 142 for each Ciprico RAID.

70. Restart GRIO to make sure that it is configured correctly:

```
# /etc/init.d/grio start
```

The system responds:

```
Total number of licensed grio streams is unlimited.
#
```

When GRIO is misconfigured it fails to start and gives a brief description of the problem.

71. When the GRIO daemon (ggd) runs, check the bandwidth for the Cipricos:

```
# grio -C -d /dev/dsk/dks3d1s7
```

The system responds with the bandwidth information:

```
GRIO information for path of  disk device /dev/dsk/dks2d2s7

GRIO information for device:
/hw/module/1/slot/io7/xbox_dualxtown/pci/0/scsi_ctlr/0/target/2/lun/
0/disk
opt i/o size:  4194304 bytes
reservations:    1 ops currently reserved (max. per quantum),   10
ops maximum
   bandwidth:    36864 kbytes/sec available,    40960 kbytes/sec
maximum

GRIO information for device:
/hw/module/1/slot/io7/xbox_dualxtown/pci/0/scsi_ctlr/0
opt i/o size:  4194304 bytes
reservations:    1 ops currently reserved (max. per quantum),   25
ops maximum
   bandwidth:    98304 kbytes/sec available,   102400 kbytes/sec
maximum
```

72. To run a secondary server that takes over VST operations when the primary server fails, you must repeat the above installation procedure for the secondary server except the steps associated with creating the index and the clip filesystems. These filesystems are shared by the two servers and only need to be created once. You must also install IRIS FailSafe, as described in Chapter 6, "Installing IRIS FailSafe."

Configuring for Booting Off a Backup Plex

The system normally boots off 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 off the secondary root plex, for example, *system.data.1.0*.

1. On the System Maintenance Menu, choose Enter Command Monitor:

```
5) Enter Command Monitor
Option? 5
Command Monitor.  Type "exit" to return to the menu.
```

2. 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.

3. Reset the SystemPartition, OSLoadPartition, and root environment variables to have 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:

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

4. Exit the Command Monitor and restart the system:

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

FlexIm License Installation Instructions

If you do not have a license for XLV, you must request one through the KEY-O-MATIC webpage: *http://www.sgi.com/Support/Licensing/*. If you request is granted, you receive a license by e-mail or in your web browser.

Once you receive the license, use the following procedure to activate it:

1. Log on to the VST server.

2. Become a super user:

   ```
   % su
   ```

3. Edit the license file.

   ```
   # vi /etc/flexlm/license.dat
   ```

4. Copy and paste the following license lines into the license file:

   ```
   FEATURE XLV sgifd 2.000 01-jan-0 0 6CC5738489148F24823C \
   HOSTID=#1761963733 vendor_info="XLV XFS PLEXING" \
   ISSUER="Silicon Graphics, Inc."
   ```

5. Save the file and exit vi.

6. Exit the root account.

   ```
   # exit
   ```

Video Server Toolkit Controls

Video Server Toolkit (VST) controls are configuration, control, and status variables implemented by VST system and device interface modules. These controls provide an easily-extensible mechanism for configuring and controlling both device-independent and device-specific features.

The controls fall into the following categories:

- "System Controls" on page 147
- "Device Controls" on page 154

System Controls

System controls are configuration variables not associated with a particular device and unit. These values can be set at startup through the use of a system-defaults file. Currently, there is no mechanism for changing the values after startup (e.g., through MVCP).

A system-defaults file resides in */usr/vtr/config/system-defaults* and is named according to the name of the subsystem.

The following subsystems and controls are currently supported:

/usr/vtr/config/system-defaults/fs

vtr.storage.fs.grio.enabled

Default: true

Enables or disables use of Guaranteed Rate I/O (GRIO) rate guarantees. Ordinarily, GRIO is used it is installed on the system and the clip being accessed resides on a real-time subvolume. This control can be used to disable all use of GRIO.

vtr.storage.fs.grio.quantum_increment

Default: 50000

Specifies the increment in microseconds used to round down GRIO time quantums for reservations. This rounding is required to ensure a manageable least-common-multiple (the time required for one I/O operation).

vtr.storage.fs.grio.reap_interval

Default: 10000000000

Specifies the interval (in nanoseconds) used for deallocating unused GRIO bandwidth reservations.

vtr.storage.fs.grio.reserve_mode {standard | maximum | average}

Specifies how much disk bandwidth is reserved for accessing a clip. The default setting for this control is *standard* which reserves a standard amount of bandwidth for the clip according to its image format and compression. The *standard* setting is intended to meet the needs of most material

The *maximum* setting reserves enough bandwidth to satisfy the largest frame in the clip. This may waste bandwidth if the largest frame is quite anomalous

The *average* setting reserves enough bandwidth to satisfy the total average bit-rate of the clip. However, this setting may not provide enough bandwidth for portions of the clip which have higher bandwidth requirements.

vtr.storage.fs.io_size

Default: 0 (not specified)

Specifies the minimum number of bytes read from or written to a clip file. Ordinarily, this control is not set and the I/O size is set equal to the major alignment.

vtr.storage.fs.latency_warning_threshold

Default: 400

Specifies the threshold (in milliseconds) which if exceeded by a single I/O operation during playback or recording will log a warning message.

vtr.storage.fs.major_alignment

Default: 0 (not specified)

Specifies the alignment boundary used to optimize access to the clip file. Ordinarily, this control is not set and the major alignment is set to the stripe size (for non-realtime subvolumes) or extent size (for realtime subvolumes) of the volume on which the clip is stored. This control can be used to override the normal major alignment system-wide.

vtr.storage.fs.max_cue_reservations

Default: 1

Specifies the maximum number of bandwidth reservations allowed per real-time filesystem for cueing clips for playback. A cueing reservation allocates one I/O operation per second. If more units are cueing than the limit specified by this control, the units share the reservation round-robin.

vtr.storage.fs.max_io_vectors

Default: 256

Specifies the maximum number of I/O vectors used in a single I/O operation.

vtr.storage.fs.minor_alignment

Default: 0 (not specified)

Specifies the alignment boundary used to match the access requirements of the clip file. Ordinarily, this control is not set and the minor alignment is set to the block size of the filesystem on which the clip is stored. This control can be used to override the normal minor alignment system-wide.

Subsystem: main (/usr/vtr/config/system-defaults/main)

vtr.main.heap.dump

Default: 0

Internal control. Do not change.

vtr.main.heap.trace_heap_pid

Default: 0

Internal control. Do not change.

vtr.main.heap.trace_pid

Default: 0

Internal control. Do not change.

vtr.main.list_threads

Default: 0

Internal control. Do not change.

vtr.main.ulock_spin_count

Default: 0

Internal control. Do not change.

vtr.main.ulock_yield_count

Default: 0

Internal control. Do not change.

vtr.main.control_heap_size

Default: 8388608

Specifies the size (in bytes) of the heap shared by all control ports. In certain system configurations, this value may need to be adjusted upwards.

vtr.main.log_level.console

Default: 0

Specifies the maximum log level of messages sent to stdout.

vtr.main.log_level.file

Default: 0

Specifies the maximum log level of messages sent to the server log file (usually, /usr/vtr/adm/logs/vtrlog).

vtr.main.log_level.syslog

Default: 0

Specifies the maximum log level of messages sent to the system log (usually */var/adm/SYSLOG*).

vtr.main.purify_new_leaks

Default: false

Internal control. Do not change.

vtr.main.thread.cpu_limit

Default: 0

Internal control. Do not change.

vtr.main.thread.cpu_limit_interval

Default: 10

Internal control. Do not change.

vtr.main.timing_standard

Default: 525

Specifies the default timing standard for the system. This default is used in certain situations when there is ambiguity about the timing standard in use. For example, certain DVCPRO DIF files may not have the timing standard information available in the headers. This control may be set to either "525" or "625" to select the default behavior.

vtr.main.unit_heap_size

Default: 8388608

Specifies the size (in bytes) of the heap allocated to each unit.

Subsystem: time (/usr/vtr/config/system-defaults/time)

Note: VST supports multiple time channels, each of which can have it's own control settings. Settings in "time" apply to all channels while settings in "time_<channel>" apply only to the specific channel.

vtr.time.slave_system_time

Default: true (for channel 1 unless another channel specifically enabled)

Specifies whether the system time (maintained by IRIX) is slaved to the timecode inputs for this time channel. Only one channel can be used to slave the system time

Depending on the stability of the incoming timecode signal, the IRIX system time usually tracks within 500us of the input timecode.

vtr.time.offset

Default: -27000000 (Miranda Little-Red), -6600000 (Horita PR-232)

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 the VST time processing.

153

Device Controls

Device controls are configuration, control, and status variables that are associated with a particular unit instance of a particular device interface. Device controls can be set at startup through the use of a *device-defaults* file. In addition, a unit's device controls can be set and queried through the MVCP GET and SET commands.

A *device-defaults* file resides in */usr/vtr/config/device-defaults device-defaults* files are loaded according to a hierarchical scheme enabling common control settings to be shared among devices.

When a unit device interface is initialized, control settings are loaded from *device-defaults* files in the following order:

- Settings for all devices: ALL
- Settings for node type: MEDIA or STORAGE
- Settings for port type: VIDEO, NETWORK, DECK, or DISK
- Settings for device class: <device-class>
- Settings for device: <device-name>

For example, when the unit device interface for the DIVO_1 video device is initialized, settings are loaded from the device-defaults files as follows:

- ALL
- MEDIA
- VIDEO
- DIVO
- DIVO_1

The following device controls are currently supported:

Device Module: VtrDQ Class: dq

vtr.deck.error

Default: (none)

This read-only control returns the last error detected from the deck control interface.

vtr.deck.status

Default: (none)

This read-only control returns the current deck transport status.

vtr.edit.coincidence.preroll

Default: 10

Specifies how far in advance of the edit in-point that VST attempts to sync. Under some circumstances, the actual coincidence point will be a few frames later than specified, so this control should be set to at least 10 frames.

vtr.edit.preroll

Default: 5:00

Specifies the deck preroll for edit operations.

vtr.edit.postroll

Default: 3:00

Specifies the deck postroll for edit operations.

vtr.edit.status

Default: (none)

This read-only control returns the current status of an edit operation that is in-progress.

vtr.media.video.frame_rate

Default: (none)

This read-only control returns the video frame rate currently being reported by the deck.

vtr.media.video.sync_port

Default: (none)

Specifies the port number from which the deck control interface should derive video sync timing. This should be set to the port that will be used for capturing from or laying-down to the deck

If the control is not set, VST will use any available port which has active video, but a warning will be issued.

vtr.media.output-mode

Default: pb

Specifies the current output mode of the deck. The value may be "pb" for normal playback or "ee" for E-to-E.

Device Module: VtrDvbVG Classes: dvb-asi

vtr.media.output.buffer_depth

Default: 0 (system determined minimum)

Specifies how much memory is used to buffer outgoing video data after it is read from the storage system. The target I/O size for the clip is multiplied by the value of this control to determine how much buffer memory is allocated

The system will allocate a minimum of 3 buffers even if the control is set to a number less than 3.

vtr.media.dvb-asi.fifo_buffer.size

Default: 65536

Specifies the parameter (fifoBufferSize) setting to the mpMuxOpen call. See *Viewgraphics Dynamo MediaPump Reference Manual* for details.

vtr.media.dvb-asi.io_buffer.size

Default: 65536

Specifies the parameter (ioBufferSize) setting to the mpMuxOpen call. See *Viewgraphics Dynamo MediaPump Reference Manual* for details.

vtr.media.dvb-asi.transfer_buffer.size

Default: 240640

Specifies the parameter (transferBufferSize) setting to the mpMuxOpen and muMuxStreamOpen call. See *Viewgraphics Dynamo MediaPump Reference Manual* for details.

vtr.media.dvb-asi.transfer_buffer.depth

Default: 6

Specifies the parameter setting (userBufferCount) to the mpMuxStreamOpen call. See *Viewgraphics Dynamo MediaPump Reference Manual* for details.

vtr.media.dvb-asi.port

Default: 0

Select which viewgraphics dvb board to use to play the loaded clip when there are multiple boards installed in the system

This control must be set before cueing and will remain in effect for that unit.

vtr.media.dvb-asi.port.total_bitrate

Default: 0

Specifies the total aggregate bit rate of the dvb-asi port. The system will default to 52000000 bits per second if this control is set to 0.

vtr.media.dvb-asi.drain_latency

Default: 4

Specifies the number of frames before the current stream finishes playing that the next stream begins cueing. This is an internal control that users should not change.

vtr.media.dvb-asi.drain_on_loop

Default: true

When set to false, the playing of subsequent passes of a loop mode play out will start immediately without waiting for the previous pass to finish. If it is set to false, there may be a discrepancy in the software-maintained frame location of the stream. This is an internal control that users should not change.

vtr.media.dvb-asi.seamless_playback

Default: false

Specifies whether to use splicing to provide a seamless concatenation of MPEG streams.

vtr.media.dvb-asi.halt_on_underflow

Default: false

Specifies the parameter (keepAlive) setting to the mpMuxOpen call. See *Viewgraphics Dynamo MediaPump Reference Manual* for details. The default is false, meaning keepAlive is true.

vtr.media.mpeg.pid.video

Default: -1 (do not remap)

This control allows the video PID of the clip played to be remapped to the one specified. The default of -1 means no remapping will be done

This control must be set before cueing and will remain in effect for that unit.

vtr.media.mpeg.pid_offset.audio

Default: 1 (video PID + 1)

This control specifies the offset of the audio PID from the video PID when the video PID is being remapped

This control must be set before cueing and will remain in effect for that unit.

vtr.media.mpeg.pid_offset.pcr {<PCR PID offset from video PID>}

Default: 0 (same as video PID)

This control specifies the offset of the PCR PID from the video PID when the video PID is being remapped

This control must be set before cueing and will remain in effect for that unit.

vtr.media.mpeg.pid.pmt

Default: -1 (do not remap)

This control specifies that the PMT (Program Map Table) PID of the clip being played is to be remapped to the specified PID. The default of -1 means no remapping will be done.

vtr.media.mpeg.bit_rate

Default: -1

Specifies the bit rate that the hardware will use to send out the clip. This control overrides the bit rate recorded in the clip. This is an internal control that users should not change.

vtr.media.mpeg.pat.frequency

Default: 30

Specifies the number of times per second the PAT (Program Association Table) is sent.

vtr.media.mpeg.pmt.frequency

Default: 30

Specifies the number of times per second the PMT (Program Map Table) is send.

vtr.media.mpeg.pcr.frequency

Default: 30

Specifies the number of times per second the PCR (Program Clock Reference) is sent.

vtr.media.mpeg.program_number

Default: -1

This control specifies that the program number of the clip being played will be remapped to the specified program number. The default of -1 means no remapping will be done.

Device Module: VtrFS Class: fs

vtr.storage.clear_after_play

Default: false

Specifies that frames are automatically cleared from the clip after being played. The clip must be loaded for input/output

This mechanism is used to implement a delay. One unit records the clip while another unit plays the same clip delayed by some fixed interval. Setting the clear_after_play control releases the disk space occupied by the frames after they have been played; hence, the disk utilization remains constant

This control takes effect immediately.

vtr.storage.continue_after_error

Default: true

Specifies that the unit continues playing after an I/O error occurs. Any frames that could not be read from the clip file will be played as black.

vtr.storage.io_log_level

Default: 4

Specifies the log level for I/O logging messages.

vtr.storage.max_open_files

Default: 8

Specifies the maximum number of clip files that the unit may have open at a single time. If the unit is playing a segmented clip with a large number of short source clips, the source clips will be opened and closed as needed to stay within the specified limit.

Device Module: VtrVela Class: vela

vtr.media.output.buffer_depth

Default: 0 (system determined minimum)

Specifies how much memory is used to buffer outgoing video data after it is read from the storage system. The target I/O size for the clip is multiplied by the value of this control to determine how much buffer memory is allocated

The system will allocate a minimum of 3 buffers even if the control is set to a number less than 3.

vtr.media.output.sync_source

Default: internal

Specifies the sync source for the video output. If set to external, genlock mode is set enabling the decoder to synchronize to the genlock input source. An external sync source should be connected to the Vela decoder. See the *Vela Research Installation and User Manual* for details.

vtr.media.output.write_size

Default: 65536

Specifies the size of the SCSI write operations used to send the MPEG stream to the decoder.

vtr.media.vela.error.retry

Default: false

Specifies whether the software should attempt an error recovery when a hardware failure is encountered.

vtr.media.vela.error.recover_realtime

Default: true

Specifies whether the software should skip over the time that hardware is not available when attempting an error recovery. If set to false, the software will start playing the clip from the point when the hardware failure occurred. If set to true, the software will restart the clip from where it should be playing as if there were no interruption.

vtr.media.vela.scsi.timeout

Default: 0

Specifies the time in seconds to wait before a SCSI timeout is detected. The default 0 means use the operating system default.

vtr.media.vela.command_latency

Default: 0

Specifies the command latency of hardware and software in frames. When a time delayed command is send to the Vela module, it will start processing the command N frames earlier than specified.

vtr.media.vela.query_decoder_location

Default: true

Specifies whether to query the Vela decoder for the current location. When set to false, the software-maintained location is used.

vtr.media.vela.seamless_playback

Default: false

Specifies whether to use splicing to provide a seamless concatenation of MPEG streams

The streams must satisfy the following constraints:

- Streams must be the same Stream type (system, program, or transport)
- Streams must have same video resolution (720x480, 352x480, 352x240 > etc)
- Video bitrate must be approximately the same
- Audio compression must be the same (bitrate, sampling rate, etc)
- Closed GOP at beginning of stream
- Streams must have the same buffering model (like STD compliant).

Often streams have too much or too little audio at the end of the stream, which causes buffer over/under flows when concatenated.

Device Module: VtrVL Classes: DIVO, DIVO_DVC, mvp

vtr.edit.postroll

Default: 3:00

Specifies the postroll duration for an edit performed by the unit. This is not the same as the postroll set for a machine control port (e.g., Sony or Louth protocol).

vtr.edit.preroll

Default: 5:00

Specifies the preroll duration for an edit performed by the unit. This is not the same as the preroll set for a machine control port (e.g., Sony or Louth protocol).

vtr.media.audio.input.channel_map {<map> | *}

Default: *

Specifies the mapping from device input audio channels to clip audio channels. The default channel map, "*", is a one-to-one mapping. A custom channel map is specified using this format: "3>1 4>2". This channel map specifies that input channel 3 should be recorded in clip channel 1, and input channel 4 should be recorded in clip channel 2

If recording is in-progress when this control is changed, the channel map will not take effect until recording is stopped and restarted.

vtr.media.audio.input.channels

Default: 2

Specifies the number of audio input channels to be read from the audio device. This can be fewer than the number of channels supported by the audio device but cannot be greater

If recording is in-progress when this control is changed, the input audio device will not be reconfigured until recording is stopped and restarted.

vtr.media.audio.input.channels.clip

Default: -1

Specifies the number of audio channels to place in a newly created clip. The default value, -1, specifies that the value will be taken from the vtr.media.audio.input.channels control. A value of 0 disables audio recording into the clip

This control is only read when a new clip is being created. Changing this control will not affect any existing clips including the one currently loaded.

vtr.media.audio.input.edge_detect

Default: 0

Enables an internal test mode. The value is the floating-point threshold value for edge detection.

vtr.media.audio.input.port

Default: DefaultIn

Specifies the name of the audio input port. If set to DefaultIn the default input audio port for the system is used. The default port can be changed through the audio control panel (apanel)

If set to VideoIn, audio input is taken from the digital audio embedded in the digital video signal (SMPTE272M)

The port may also be specified using the form:

[<subsystem>.]<device>.<interface>

For audio ports based upon the Radical audio subsystem, the subsystem is "RAn"; the device is one of "AnalogIn", "AESIn", or "ADATIn"; the interface is "LineIn" or "Microphone" for the analog device, and "AESInput" or "OpticalIn" for the AES device

For example, to take audio input for the DIVO_1 video 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.AESInput"

If recording is in-progress when this control is set, the audio port will not be reconfigured until recording has been stopped and restarted.

vtr.media.audio.input.rate

Default: 48000

Specifies the audio sampling rate in samples per second

If recording is in-progress when this control is set, the audio port will not be reconfigured until recording has been stopped and restarted.

vtr.media.audio.input.sample.format

Default: twos-complement

Specifies the format of the audio samples. The choices are "twos- complement", "float", and "double"

If recording is in-progress when this control is set, the audio port will not be reconfigured until recording has been stopped and restarted.

vtr.media.audio.input.sample.width

Default: 24

Specifies the number of bits of precision in each audio sample. Valid values are 8, 16, and 24. Note that the 24-bit setting uses 4 bytes (32 bits) to store each 24-bit audio sample

If recording is in-progress when this control is set, the audio port will not be reconfigured until recording has been stopped and restarted.

vtr.media.audio.input.skew

Default: 0

Specifies a skew value (in nanoseconds) for adjusting the synchronization between the input audio and video streams. The skew control can be used to compensate for a small amount (< 1 frame time) of excess delay on either the audio or video path into the system

The skew value takes effect the next time the audio and video streams are synchronized which is the next time recording is started or resumed after pausing.

vtr.media.audio.input.sync_source

Default: <blank> (Audio clock unchanged)

Specifies the sync source to be used for clocking the input audio. The sync source may only be changed for analog audio inputs. Digital audio inputs are self-clocking

A value of "Internal" specifies that the audio subsystem's internal oscillator should be used. A value of "Video" specifies that the external video reference signal should be used as the sync source. A value of "Internal Video" specifies that the internal video synchronization signal (cosmic) should be used

The sync source may also be specified using the form:

[<subsystem>.]<clock name>

For audio ports based upon the Radical audio subsystem, the subsystem is "RAn", and the clock name is one of "48 kHz Internal Clock", "44100 Hz Internal Clock", "AES Clock", "ADAT Clock", "48 kHz Video Clock", or "44100 Hz Video Clock"

If recording is in-progress when this control is set, the audio port will not be reconfigured until recording has been stopped and restarted.

vtr.media.audio.output.channel_map {<map> | *}

Default: *

Specifies the mapping from clip audio channels to device output audio channels. The default channel map, "*", is a one-to-one mapping. A custom channel map is specified using this format: "3>1 4>2". This channel map specifies that clip channel 3 should be sent out on audio channel 1, and clip channel 4 should be sent out on audio channel 2

If playback is in-progress when this control is changed, the channel map will not take effect until playback is stopped and restarted.

vtr.media.audio.output.channels

Default: 2

Specifies the number of audio output channels to be written to the audio device. This can be fewer than the number of channels supported by the audio device but cannot be greater

If playback is in-progress when this control is changed, the output audio device will not be reconfigured until playback is stopped and restarted.

vtr.media.audio.output.fade.duration.in

Default: 10000000

Specifies the time (in nanoseconds) used for fading-in the audio at a clip transition point. Certain clip formats automatically mark transitions at the in- and out-points of an insert edit. Other formats support transitions only when starting or stopping clip playback.

vtr.media.audio.output.fade.duration.out

Default: 10000000

Specifies the time (in nanoseconds) used for fading-out the audio at a clip transition point. Certain clip formats automatically mark transitions at the in- and out-points of an insert edit. Other formats support transitions only when starting or stopping clip playback.

vtr.media.audio.output.port

Default: DefaultOut

Specifies the name of the audio output port. If set to DefaultOut the default output audio port for the system is used. The default port can be changed through the audio control panel (apanel)

If set to VideoOut, audio output is sent out as digital audio embedded in the digital video signal (SMPTE272M)

The port may also be specified using the form:

[<subsystem>.]<device>.<interface>

For audio ports based upon the Radical audio subsystem, the subsystem is "RAn"; the device is one of "AnalogOut", "AESOut", or "ADATOut"; the interface is "LineOut" for the analog device, and "AESOutput" or "OpticalOut" for the AES device

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

vtr.media.audio.output.port "RAD1.AESOut.AESOutput"

If playback is in-progress when this control is set, the audio port will not be reconfigured until playback has been stopped and restarted.

vtr.media.audio.output.rate

Default: 48000

Specifies the audio sampling rate in samples per second

This control is used to select the audio rate for tone output if no clip is loaded. The audio parameters of the clip are used for normal playback. The input audio parameters are used to configure end-to- end mode

If playback is in-progress when this control is set, the audio port will not be reconfigured until playback has been stopped and restarted.

vtr.media.audio.output.sample.format

Default: twos-complement

Specifies the format of the audio samples. The choices are "twos- complement", "float", and "double"

This control is used to select the audio sample format for tone output if no clip is loaded. The audio parameters of the clip are used for normal playback. The input audio parameters are used to configure end-to-end mode

If playback is in-progress when this control is set, the audio port will not be reconfigured until playback has been stopped and restarted.

vtr.media.audio.output.sample.width

Default: 24

Specifies the number of bits of precision in each audio sample. Valid values are 8, 16, and 24. Note that the 24-bit setting uses 4 bytes (32 bits) to store each 24-bit audio sample

This control is used to select the audio sample precision for tone output if no clip is loaded. The audio parameters of the clip are used for normal playback. The input audio parameters are used to configure end-to-end mode

If playback is in-progress when this control is set, the audio port will not be reconfigured until playback has been stopped and restarted.

vtr.media.audio.output.skew

Default: 0

Specifies a skew value (in nanoseconds) for adjusting the synchronization between the output audio and video streams. The skew control can be used to compensate for a small amount (< 1 frame time) of excess delay on either the audio or video path after leaving the system

The skew value takes effect the next time the audio and video streams are synchronized which is the next time playback is started or resumed after pausing.

vtr.media.audio.output.sync_source

Default: Video

Specifies the sync source to be used for clocking the output audio. The sync source may only be changed for both digital and analog audio outputs

A value of "Internal" specifies that the audio subsystem's internal oscillator should be used. A value of "Video" specifies that the external video reference signal should be used as the sync source. A value of "Internal Video" specifies that the internal video synchronization signal (ksync) should be used

The sync source may also be specified using the form:

[<subsystem>.]<clock name>

For audio ports based upon the Radical audio subsystem, the subsystem is "RAn", and the clock name is one of "48 kHz Internal Clock", "44100 Hz Internal Clock", "AES Clock", "ADAT Clock", "48 kHz Video Clock", or "44100 Hz Video Clock"

If playback is in-progress when this control is set, the audio port will not be reconfigured until playback has been stopped and restarted.

vtr.media.audio.output.tone.frequency

Default: 1000

Specifies the frequency (in Hz) of the tone output when the unit output mode is set to "image"

If tone is being output when this control is set, the frequency will not be changed until tone (output mode) is changed.

vtr.media.audio.output.tone.level

Default: -20

Specifies the level (in dB) of the tone output when the unit output mode is set to "image"

If tone is being output when this control is set, the level will not be changed until tone (output mode) is changed.

vtr.media.clip.format

Default: movie/vframe

Specifies the format of new clips that are created This control is only read when a new clip is being created. Changing this control will not affect any existing clips including the one currently loaded.

vtr.media.clip.limit.enabled

Default: true

Specifies whether any starting or ending limits are enforced when playing back the clip. When clip limits are disabled, the clip location is free to take on any value. Frames that are not present in the clip are displayed as black (and silent).

vtr.media.clip.limit.end

Default: *

Specifies the end (upper) limit beyond which the unit is not allowed to go. The default value, "*", specifies that the limit follows the limit of recorded material in the clip. That is, the limit is the last recorded frame in the clip

This control takes effect the next time a clip is cued.

vtr.media.clip.limit.start

Default: *

Specifies the start (lower) limit beyond which the unit is not allowed to go. The default value, "*", specifies that the limit follows the limit of recorded material in the clip. That is, the limit is the first recorded frame in the clip

This control takes effect the next time a clip is cued.

vtr.media.clip.location.preset

Default: * (none)

Specifies the location to which the clip should be cued if playback or recording is started without manually cueing the unit (CUE or CUER). If this control is not set, the unit is cued to the last location played (or recorded).

vtr.media.clip.segmented.format

Default: <blank> (none)

Specifies the format to use for creating segmented clips. If this control is set when a new clip is created, the new clip is created using the segmenting format specified. The segments of the clip are created using the format specified by vtr.media.clip.format.

vtr.media.clip.segmented.insert_mode

Default: overwrite

Specifies whether a new segment that is recorded into a segmented clip overwrites the existing frames in the clip or is inserted into the clip at the cued location. The value may be "overwrite" or "insert".

vtr.media.clip.start.mode

Default: preset

Specifies the mode for determining the timecode of the first frame recorded into an empty clip if no timecode is specified when the unit is cued for recording. A value of "preset" specifies that the timecode is taken from vtr.media.clip.start.preset. A value of "time-of-day" specifies that the timecode is taken from the current time-of-day when the first frame comes into the system. A value of "vitc" specifies that the timecode is taken from the VITC embedded in the first frame

This control is only read when a newly created clip is cued for recording and no in-point has been specified for cueing.

vtr.media.clip.start.preset

Default: 01:00:00:00

Specifies the timecode of the first frame recorded into an empty clip if no timecode is specified when the unit is cued for recording and the vtr.media.clip.start.mode is set to "preset".

This control is only read when a newly created clip is cued for recording and no in-point has been specified for cueing.

vtr.media.ee.offset_frames

Default: 5

Specifies the offset (in video frames) between the input video/audio and the output video/audio when E-to-E mode is active. The minimum offset (delay) available depends upon the type of video compression being used. Uncompressed, Rice, and JPEG video have a minimum of 3 frames of offset. DVCPRO and DV require 5 frames.

vtr.media.ee_follows_record

Default: false

Not supported.

vtr.media.graphics.enabled

Default: true

Specifies whether the video output is displayed in a window on the graphical display. Currently, this control is supported only O2 systems

This control is read only during port initialization which occurs only at system startup.

vtr.media.graphics.icon_name

Default: Video Server Toolkit

Specifies the icon name associated with the graphical display window. Currently, this control is supported only O2 systems

This control is read only during port initialization which occurs only at system startup.

vtr.media.graphics.window_name

Default: Video Server Toolkit

Specifies the window name associated with the graphical display window. Currently, this control is supported only O2 systems

This control is read only during port initialization which occurs only at system startup.

vtr.media.input.abort_on_dropped_frame

Default: false

Specifies whether a Record operation will be automatically aborted if an input frame is dropped (missed). This control may be set to ensure that a Record free of errors.

vtr.media.input.buffer_depth

Default: 3

Specifies how much memory is used to buffer incoming video data before it is written to the storage system. The target I/O size for the clip is multiplied by the value of this control to determine how much buffer memory is allocated

If video compression is being used, additional buffer memory may be automatically allocated to ensure that the target I/O size can be met even during extremely compressible segments of an input program

If recording is in-progress when this control is changed, the video buffer memory will not be reconfigured until recording is stopped and restarted.

vtr.media.input.frame_log_level

Default: 4

Specifies the logging level for input frame debugging messages.

vtr.media.input.idle_mode

Default: off

Specifies the idle-mode behavior for the input port. If the value is set to "discard", the input port is enabled and frames are continuously read and discarded from the port

Idle-mode discard can be used to shorten the initialization time for beginning a record operation. It also allows command synchronization using the input video timing

This feature may be revised in a future release.

vtr.media.input.mode

Default: store

Specifies the recording behavior for the input port. If the value is set to "discard", the frames read from the input port are discarded instead of being recorded. However, the clip location continues to advance at the normal rate.

vtr.media.input.preemptible

Default: true

Specifies whether the unit may be preempted while idle if the idle- mode is something other than "off". Setting this control to false prevents another unit from gaining control of the input port

This feature may be revised in a future release.

vtr.media.input.timecode.preset

Default: 0

Specifies the initial value of the free-running counter which is used by the VITC and userbits free-running mode. Also specifies a new starting value for the running counter

This control takes effect immediately.

vtr.media.input.trigger.mode.in

Default: none

Specifies the type of trigger used to begin recording. The default value, "none", disables triggering; recording begins immediately (subject to command scheduling if the record command is time- stamped)

The value "vitc" enables VITC-triggering. When the unit record command is received, the unit begins monitoring the VITC of each input frame. When an input frame whose VITC matches the value of vtr.media.input.trigger.vitc.in is received, recording begins. The triggering frame is the first frame recorded to the clip

This control is read each time an input frame is received.

vtr.media.input.trigger.mode.out

Default: none

Specifies the type of trigger used to end recording. The default value, "none", disables triggering; recording ends immediately after the unit receives a stop command (subject to command scheduling if the record command is time-stamped)

The value "vitc" enables VITC-triggering. When recording begins, the unit begins monitoring the VITC of each input frame as it is recorded. When an input frame whose VITC matches the value of vtr.media.input.trigger.vitc.in is received, recording ends. The triggering frame is *not* recorded

This control is read each time an input frame is received.

vtr.media.input.trigger.vitc.in

Default:

Specifies the VITC of the first frame recorded when the input trigger-in mode is VITC

This control is read each time a input frame is received.

vtr.media.input.trigger.vitc.out

Default:

Specifies the VITC of the frame following the last frame recorded when the input trigger-out mode is VITC

This control is read each time a input frame is received.

vtr.media.input.userbits.mode

Default: source

Specifies the data stored in the timecode userbits for each recorded frame. The possible values are:

- off   Zeroed
- source   Userbits from input source
- freerun   Value of free-running counter
- run   Value of running counter
- date   Current date
- time   Current (local) time
- constant   Value of B<vtr.media.input.userbits.preset>

The free-running counter begins incrementing immediately from the value set into vtr.media.input.userbits.preset and continues running regardless of whether the unit is recording or idle

The running counter increments from the value set into vtr.media.input.userbits.preset, but only when the unit is actively recording

This control takes effect immediately.

vtr.media.input.userbits.preset

Default: 0

Specifies the initial value of the free-running and running userbits counters. Also specifies the constant value stored in the userbits if the userbits mode is "constant"

This control takes effect immediately. The free-running counter begins incrementing immediately. The running counter begins incrementing if the unit is recording.

vtr.media.output.buffer_depth

Default: 3

Specifies how much memory is used to buffer outgoing video data after it is read from the storage system. The target I/O size for the clip is multiplied by the value of this control to determine how much buffer memory is allocated

If video compression is being used, additional buffer memory may be automatically allocated (on some platforms) to ensure that the target I/O size can be met even during extremely compressible segments of a clip

If playback is in-progress when this control is changed, the video buffer memory will not be reconfigured until playback is stopped and restarted.

vtr.media.output.cue_buffer_depth

Default: 1

Specifies how much memory is used to buffer outgoing video data for a clip that is being cue to play after it is read from the storage system. The target I/O size for the clip is multiplied by the value of this control to determine how much buffer memory is allocated for cueing

A separate buffer pool is maintained for cueing to ensure timely completion of cueing when a clip is currently playing on the same media port from a different unit

If video compression is being used, additional buffer memory may be automatically allocated (on some platforms) to ensure that the target I/O size can be met even during extremely compressible segments of a clip

If playback is in-progress when this control is changed, the video buffer memory will not be reconfigured until playback is stopped and restarted.

vtr.media.output.cued_mode

Default: output

Specifies the unit's output mode when it is cued. The possible values are:

- output   Use the value of vtr.media.output.mode
- idle   Use the value of vtr.media.output.idle_mode
- clip   Normal clip video/audio output
- hold   Freeze on last image (silent audio)
- black   Black (silent audio)
- image   Still image (vtr.media.video.output.image.type)
- ee   End-to-end (input video/audio)

vtr.media.output.idle_mode

Default: hold

Specifies the unit's output mode when it is not playing. The possible values are:

- off   No output (port stopped)
- hold   Freeze on last image (silent audio)
- black   Black (silent audio)
- image   Still image (vtr.media.video.output.image.type)
- ee   End-to-end (input video/audio)

vtr.media.output.mode

Default: clip

Specifies the unit's output mode when it is playing. The possible values are:

- clip   Normal clip video/audio output
- hold   Freeze current image (silent audio)
- black   Black (silent audio)
- image   Still image (vtr.media.video.output.image.type)
- ee   End-to-end (input video/audio)

Regardless of the output mode, the clip location continues to advance.

vtr.media.output.pause_at_limits

Default: false

Specifies whether the unit should pause (instead of stop) when a start or end limit is reached.

vtr.media.output.preemptible

Default: true

Specifies whether the unit may be preempted while idle if the idle- mode is something other than "off". Setting this control to false prevents another unit from gaining control of the output port

This feature may be revised in a future release.

vtr.media.output.speed.fast_forward

Default: 50000

Specifies the speed of playback when a FastForward (FF) command is issued (1000 = 1.0x).

vtr.media.output.speed.rewind

Default: -50000

Specifies the speed of playback when a Rewind (REW) command is issued (1000 = 1.0x).

vtr.media.output.timecode.preset

Default: 0

Specifies the initial value of the free-running counter which is used by the VITC and userbits free-running mode. Also specifies a new starting value for the running counter

This control takes effect immediately.

vtr.media.output.userbits.mode

Default: source

Specifies the data sent in the timecode userbits of each frame. The possible values are:

- off   Zeroed
- source   Userbits from recorded clip
- freerun   Value of free-running counter
- run   Value of running counter
- date   Current date
- time   Current (local) time
- constant   Value of vtr.media.output.userbits.preset

The free-running counter begins incrementing immediately from the value set into vtr.media.output.userbits.preset and continues running regardless of whether the unit is playing or idle

The running counter increments from the value set into vtr.media.output.userbits.preset, but only when the unit is actively playing

This control takes effect immediately.

vtr.media.output.userbits.preset

Default: 0

Specifies the initial value of the free-running and running userbits counters. Also specifies the constant value stored in the userbits if the userbits mode is "constant"

This control takes effect immediately. The free-running counter begins incrementing immediately. The running counter beings incrementing if the unit is playing.

vtr.media.port.enabled

Default: true

Specifies whether this media port is enabled for use by VST applications. This control can be used to selectively disable media ports which may be used by other applications

This control is read only during port initialization which occurs only at system startup. Once VST has started, a port cannot be disabled or enabled.

vtr.media.port.shared_access

Default: true

Specifies whether access to this media port is shared between VCP- Recorder and other applications. VST attempts to gain exclusive control of the port when a unit is cued for play or record on the port and releases control of the port when the unit is stopped

This control is read only during port initialization which occurs only at system startup.

vtr.media.port.use_fillpoints

Default: auto

Internal control. Do not change.

vtr.media.video.beep_on_dropped_frame

Default: no

Specifies whether a bell character (\007) is written to the log when a frame is dropped. This control is primarily for debugging

This control takes effect immediately.

vtr.media.video.input.bit_rate.feedback_gain

Default: 0.01

Specifies the magnitude of the feedback applied in adjusting the compression quality when recording a clip using a lossy compression method with a target bit rate. A larger value causes the compression quality adjustment to be larger when the size of a compressed frame deviates from the target size

Because there is a delay through the compressor, if the feedback is too large, the compression stage will become unstable resulting in wild oscillations in compression ratio/bit rate

This control takes effect immediately.

vtr.media.video.input.bit_rate.target

Default: 25000000

Specifies the target bit rate for the compression stage when a clip is being recorded using a lossy compression method (e.g., JPEG). The compression quality will be automatically adjusted in response to variations in the size of the compressed video frames as the input material becomes more or less compressible

Specify a value of 0 to disable automatic compression quality adjustment. The compression quality will be locked to the value specified by vtr.media.video.input.compression.quality

This control takes effect immediately.

vtr.media.video.input.colorspace

Default: ccir601

Specifies the colorspace encoding used for recording the input video. The possible values are:

- ccir601   Recommendation 601, YCbCr
- yuv   Full-range YCbCr rp175 RP-175 RGB
- rgb   Full-range RGB

This control is only read when a new clip is being created. Changing this control will not affect any existing clips including the one currently loaded.

vtr.media.video.input.compression.dithering

Default: off

This control is not currently implemented.

vtr.media.video.input.compression.precision

Default: 8

Specifies the number of bits of precision used by the compression algorithm to store the video component data

If the compression type is "rice", the supported values are 8, 10, 12, and 13

This control is only read when a new clip is being created. Changing this control will not affect any existing clips including the one currently loaded.

vtr.media.video.input.compression.quality

Default: 0.94

Specifies the value of the quality parameter supplied to the lossy compression algorithm. The value is specified from 0.0 to 1.0

For the JPEG compressor, the quality is a rough indication of image detail preservation. 0.99 is nearly perfect

If vtr.media.input.bit_rate.target is non-zero, the quality will be adjusted after each frame is compressed. If this control is changed while recording is in progress, the quality will be set to the specified value for the next frame to be compressed, but automatic rate-control adjustment will resume after the next frame exits the compression stage

This control takes effect immediately.

vtr.media.video.input.compression.sampling

Default: 422

Specifies the type of video component sampling used by the compression algorithm

If the compression type is "rice", the supported values are:

- 422   4:2:2 YCbCr
- 4224   4:2:2:4 YCbCrA
- 444   4:4:4
- 4444   4:4:4:4

vtr.media.video.input.compression.type

Default: default

Specifies the type of compression applied to video frames being recorded

On Origin with the XT-DIVO video board, the possible values are:

- none   No compression (except for component/precision subsampling)
- rice   Lossless entropy-coding via DIVO hardware
- dvcpro   DVCPRO compression via DIVO hardware
- dv   DV compression via DIVO hardware

If "default" is specified on this platform, "rice" is automatically selected

On O2 with the O2Video (mvp) video subsystem, the possible values are:

- none   No compression (except for component/precision subsampling)
- jpeg   Intra-field JPEG compression via ICE hardware

If "default" is specified on this platform, "jpeg" is automatically selected

This control is only read when a new clip is being created. Changing this control will not affect any existing clips including the one currently loaded.

vtr.media.video.input.field_dominance

Default: f1

Specifies the input field dominance. Possible values are f1 and f2

This control is only read when a new clip is being created. Changing this control will not affect any existing clips including the one currently loaded.

vtr.media.video.input.format

Default: normal

Specifies the format of the input video data

On Origin with the XT-DIVO video board, the possible values are:

- normal   Single-link SDI (CCIR601) input
- digital-component-serial   (same)
- sdi   (same)
- digital-component-dual-serial Dual-link SDI (CCIR601) input
- sdi-dual-link   (same)
- digital-component-rgb-serial   Dual-link RGB (RP-175) input
- sdti   Serial Data Transport Interface

On O2 with the O2Video (mvp) video subsystem, the possible values are:

- normal
- composite   Analog composite input
- svideo   Analog S-Video input
- digital-camera   Digital camera input

This control is only read when a new clip is being created. Changing this control will not affect any existing clips including the one currently loaded.

vtr.media.video.input.frame_mode

Default: false

Specifies whether the input video fields are stored "frames" for compression modes that support either mode. Rice compression supports only field-mode while DVCPRO supports only frame-mode.

vtr.media.video.input.hard_reset

Default: false (edge-triggered)

When this control is set to true, the associated hardware input port is reset. (Supported only for DIVO).

vtr.media.video.input.packing

Default: R242_8

Specifies the video component packing

On Origin with the XT-DIVO video board, the possible values are:

- 444_332
- R444_332
- 0444_8
- R0444_8
- 444_8
- R444_8
- 4444_8
- R4444_8
- 242_8
- R242_8
- 4_8
- 4_4
- 4444_10_10_10_2
- 2424_10_10_10_2Z
- R2424_10_10_10_2Z
- 242_10_in_16_L
- 242_10_in_16_R

- 4444_10_in_16_L

- 4444_10_in_16_R

- 4444_13_in_16_L

- 4444_13_in_16_R

- 4444_12_in_16_L

- 4444_12_in_16_R

- X4444_5551

- 4444_12

- 444_12

- 4444_6

- 444_5_6_5

- 242_10

- 444_10_in_16_L

- R242_10_in_16_L

- R242_10_in_16_R

- R242_10

On O2 with the O2Video (mvp) video subsystem, the possible values are:

- RGBA_8

- RGB_8

- YVYU_422_8

- ABGR_8

- YVYU_422_10

- AUYV_4444_8

- ARGB_1555

vtr.media.video.input.restart_attempts

Default: 4

Specifies the number of times that VST will attempt to restart a failed input video transfer (recording) before giving up and raising an error on the unit

This control takes effect immediately.

vtr.media.video.input.source

Default: normal

Specifies the source for input video

On Origin with the XT-DIVO video board, the possible values are:

- normal   Single- or dual-link Silicon Graphics inputs
- loopback   Video from output looped back to input

This control takes effect the next time the unit is cued for recording.

vtr.media.video.input.timing

Default: auto

Specifies the timing for input video. Possible values are:

- auto   Timing read from current video input
- 525   525/59.94 timing
- 625   625/50 timing

Note: Input timing cannot be set on Origin/DIVO. The input timing is always taken from the video input currently connected

This control is only read when a new clip is being created. Changing this control will not affect any existing clips including the one currently loaded.

vtr.media.video.input.vitc.mode

Default: source

Specifies the mode used for storing VITC for frames as they are recorded into a clip. Possible values are:

- off   Zeroed
- source   VITC from video input
- clip   Clip location (CTL) timecode
- freerun   Value of free-running counter
- run   Value of running counter

The free-running counter begins incrementing immediately from the value set into vtr.media.input.timecode.preset and continues running regardless of whether the unit is recording or idle

The running counter increments from the value set into vtr.media.input.timecode.preset, but only when the unit is actively recording

This control takes effect immediately.

vtr.media.video.input.vitc_line_offset

Default: 14

Specifies the video line used to obtain VITC from the input video signal

This control takes effect the next time the unit is cued for recording.

vtr.media.video.output.fast_shuttle_repeat_count

Default: 2

Specifies the number of times each displayed frame is repeated when playing forward or backward at a speed greater than normal playback speed. Frames are repeated to limit the I/O bandwidth used on the clip cache filesystem when playing a high speeds

A value of 2 specifies that each displayed frame is repeated twice. For example, if playing at 2x normal playback speed, frame 1 is displayed twice, then frame 5 (not 3) is displayed twice, an so on

This control takes effect immediately. However, if this control is changed while fast playback is in progress, it may disturb the playback rate for a very brief period while the adjustment makes its way through the disk/video buffering mechanism.

vtr.media.video.output.field_dominance

Default: f1

Specifies the output video field dominance. The possible values are f1 and f2

This control is only used when starting end-to-end mode and no clip is loaded. Otherwise, the field dominance of the clip is always used to select the output field dominance

This control takes effect the next time end-to-end mode is entered from an idle state with no clip loaded.

vtr.media.video.output.format

Default: normal

Specifies the format of the output video data

On Origin with the XT-DIVO video board, the possible values are:

- normal   Single-link SDI (CCIR601) output
- digital-component-serial   (same)
- sdi   (same)
- digital-component-dual-serial Dual-link SDI (CCIR601) output
- sdi-dual-link   (same)
- digital-component-rgb-serial   Dual-link RGB (RP-175) output
- sdti   Serial Data Transport Interface

On O2 with the O2Video (mvp) video subsystem, the possible values are: normal.

This control takes effect the next time the unit is cued for playback.

vtr.media.video.output.hard_reset

Default: false (edge-triggered)

When this control is set to true, the associated hardware output port is reset. (Supported only for DIVO).

vtr.media.video.output.image.name

Default: black

Specifies the name of the image displayed when the unit output mode is "image" and the vtr.media.output.image.type is "user". The image must be placed in the appropriate directories under */usr/vtr/data/images*.

vtr.media.video.output.image.type

Default: bars

Specifies the image displayed when the unit output mode is "image". The possible values are:

- smpte-75-bars   75% saturation SMPTE color bars with PLUGE

- 75-bars-over-red   75% saturation color bars over red

- bars   smpte-75-bars (525), 75-bars-over-red (625)

- black   Black

- user   User (vtr.media.video.output.image.name)

vtr.media.video.output.max_queued_frames

Default: 1

Specifies the maximum number of video frames that are queued for video output at any time. A small number 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, Pause

This control takes effect the next time the unit is cued for playback.

vtr.media.video.output.overlay.mode

Default: fast

Specifies the whether the text overlay should be done with as little CPU overhead as possible ("fast") or should be done so that the output text is always "clean"

If this control is set to "fast" and frames are repeated either because of speed adjustments or pausing, the overlay text will be written over the existing text. If this control is set to "clean", each repeated frame is copied so that the overlay will be drawn cleanly in each frame.

vtr.media.video.output.overlay.offset.horizontal

Default: 64

Specifies the horizontal pixel offset of the left edge of the text overlay. A positive number specifies an offset from the left edge of the image while a negative number specifies an offset from the right edge

This control takes effect immediately.

vtr.media.video.output.overlay.offset.vertical

Default: -64

Specifies the horizontal pixel offset of the upper edge of the text overlay. A positive number specifies an offset from the top edge of the image while a negative number specifies an offset from the bottom edge

This control takes effect immediately.

vtr.media.video.output.overlay.text

Default: <none>

Specifies the text of the overlay drawn into each outgoing video field. The overlay is not available when playing video which is decompressed by the video board (e.g., DIVO/Rice, DIVO/DVCPRO). The overlay is also not available when the unit is playing black, is paused, or in end-to-end mode while recording

The string "\n" may be embedded in the text string to signify a line-break

The following substitution variables may also be included:

- $br   Bit-rate, instantaneous
- $br1min   Bit-rate, 1-minute rolling average
- $br1sec   Bit-rate, 1-second rolling average
- $brclip   Bit-rate, cumulative clip

- $brport   Bit-rate, cumulative port
- $clip   Clip name
- $date   Current date
- $cr   Compression ratio, instantaneous
- $cr1min   Compression ratio, 1-minute rolling average
- $cr1sec   Compression ratio, 1-second rolling average
- $crclip   Compression ratio, cumulative clip
- $crport   Compression ratio, cumulative port
- $frame   Current clip frame number
- $speed   Unit speed
- $tc   Current clip timecode
- $time   Current time (HH:MM:SS:FF)
- $vitc   VITC
- $ub   VITC userbits
- $srctc   Time code from source clip (in segmented clip)
- $src   Name of source clip (in segmented clip)
- $ust   UST when frame leaves output jack

vtr.media.video.output.repeat_both_fields

Default: yes

Specifies whether both fields of a video frame are displayed when a frame is repeated. Frames are repeated when the unit is paused or when playing a speed greater or less than normal speed. If this control is set to "no", the first field is displayed twice (line- doubled)

Repeating both fields of the frame results in better spatial resolution but will result in flicker if there is significant movement in the image between the fields

This control takes effect immediately.

vtr.media.video.output.restart_attempts

Default: 4

Specifies the number of times that VST will attempt to restart a failed output video transfer (playback) before giving up and raising an error on the unit

This control takes effect immediately.

vtr.media.video.output.skip_on_drop

Default: true

Specifies when one or more frames are dropped (missed) during playback whether the same number of frames are automatically skipped in the clip so that the clip plays in real-time.

vtr.media.video.output.sync_source

Default: external

Specifies the source of synchronization for video output

On Origin with the XT-DIVO video board, the possible values are:

- internal   Internal clock
- external   External genlock (GEN-IN) video ref signal
- digital-input-link-a External genlock to digital input link A
- digital-input-link-b External genlock to digital input link B

On O2 with the O2Video (mvp) video subsystem, the possible values are:

- internal   Internal clock
- external   External genlock video reference signal

vtr.media.video.output.timing

Default: 525

Specifies the timing standard for video output. This control is only used when entering image output mode (e.g., bars) and no clip is loaded. Possible values are:

- 525   525/59.94 timing
- 625   625/50 timing

This control takes effect the next time image output mode is entered from an idle state with no clip loaded.

vtr.media.video.output.vitc.line_offset

Default: 14

Specifies the video line where the VITC is placed in the output signal

This control takes effect the next time the unit is cued for playback.

vtr.media.video.output.vitc.mode

Default: clip

Specifies the data sent for the VITC of each frame. The possible values are:

- off   Zeroed
- source   VITC from recorded clip
- clip   Clip location (CTL) timecode
- freerun   Value of free-running counter
- run   Value of running counter

The free-running counter begins incrementing immediately from the value set into vtr.media.output.timecode.preset and continues running regardless of whether the unit is playing or idle

The running counter increments from the value set into vtr.media.output.timecode.preset, but only when the unit is actively playing

This control takes effect immediately.

Device Module: VtrVLAN Class: vlan

vtr.deck.error

Default: (none)

This read-only control returns the last error detected from the deck control interface.

vtr.deck.status

Default: (none)

This read-only control returns the current deck transport status.

vtr.edit.coincidence.preroll

Default: :10

Specifies how far in advance of the edit in-point that VST attempts to sync. Under some circumstances, the actual coincidence point will be a few frames later than specified, so this control should be set to at least 10 frames.

vtr.edit.preroll

Default: 5:00

Specifies the deck preroll for edit operations.

vtr.edit.postroll

Default: 3:00

Specifies the deck postroll for edit operations.

vtr.edit.status

Default: (none)

This read-only control returns the current status of an edit operation that is in-progress.

vtr.media.video.frame_rate

Default: (none)

This read-only control returns the video frame rate currently being reported by the deck.

vtr.media.output-mode

Default: pb

Specifies the current output mode of the deck. The value may be "pb" for normal playback or "ee" for E-to-E.

Device Module: VtrOdetics Classes: odetics, sony

vtr.control.clip.name

Default: (none)

Specifies the name of a clip to load on a Sony-protocol-controlled port. If the clip does not exist, it is created first.

vtr.control.default_id.name

Default: (none)

Specifies the name of the default id (clip) to be loaded on an Odetics-protocol-controlled port. If the clip does not exist, it is created first.

vtr.control.description

Default: (none)

Specifies a descriptive name for the control port such as "Recorder" or "Playback Deck A" or "On-Air B".

vtr.control.device_type_id

Default: 0xd803

Specifies the device type identifier returned in response to a DEVICE-TYPE protocol request.

vtr.control.edit.delay

Default: 5

Specifies the delay after an EDIT_ON or EDIT_OFF command is received before the specified operation commences.

vtr.control.ee.delay

Default: 5

Specifies the delay after an EE_ON or EE_OFF command is received before the specified operation commences.

vtr.control.ee.mode

Default: player

Specifies whether the video output is delayed so that seamless playback/EE switching occurs ("recorder" mode) or whether the video output matches the timecode reported by VST ("player" mode).

vtr.control.ee.record_select

Default: true

Specifies whether the output is switched to E-to-E mode when recording.

vtr.control.output.idle_mode

Default: hold

Specifies the mode of the output port when no playback is occuring. (See vtr.media.output.idle_mode for the video device module).

vtr.control.preread

Default: false

Unsupported.

vtr.control.sony.input_discard

Default: true

Specifies whether the input port is always active and discarding frames when no recording is occuring. If this control is set to false, the port is available for playback only.

vtr.control.sony.log_level

Default: 3

Specifies the log level of control processor debugging messages.

vtr.control.sony.timecode_log_level

Default: 0

Specifies the log level of control processor timecode debugging messages.

vtr.control.superimpose.enabled

Default: false

Specifies whether the superimpose display is enabled. This control is not effective when using certain video compression modes.

vtr.control.timecode.mode

Default: drop-frame

Specifies whether the control port reports "drop-frame" or "non- drop-frame" timecode for 525/59.94 video.

vtr.edit.postroll

Default: 3:00

Specifies the postroll for an AUTOEDIT operation.

vtr.edit.preroll

Default: 5:00

Specifies the preroll for an AUTOEDIT operation.

Automating Video Server Toolkit Commands Using Perl

You can implement Video Server Toolkit commands in Perl scripts to automate their use for such things as testing. This appendix discusses the */usr/vtr/lib/perl/MVCP.pm* Perl module, which automates back-to-back recording. The module uses the following arguments;

```
% MVCP.pm host port number duration
```

host, is the name of the host used in the telnet connection.

port is the name of the port connected to the unit.

number is the number of times the sequence, load, cue, and record, is repeated. The default is one. The argument is optional.

duration is the duration of the recording. The default is ten seconds. The argument is optional.

The *MVCP.pm* Perl script performs the following functions:

1. Connect to the host.

2. Create two units.

3. Load, cue, synchronize both units, and records media.

Example C-1   Back-to-Back Recording Using Perl

```
BEGIN {
    my $ROOT = $ENV{'ROOT'};
    if (!$ROOT) {
$ROOT = '';
    }
    unshift(@INC,"$ROOT/usr/vtr/lib/perl");
}
```

```
require MVCP;

my $vtr = Vtr::connect($ARGV[0]);
if (!$vtr) {
    exit 1;
}

my $u1 = $vtr->createUnit($ARGV[1]);
if (!$u1) {
    exit 1;
}

my $u2 = $vtr->createUnit($ARGV[1]);
if (!$u2) {
    exit 1;
}

my $num = $ARGV[2];
if (!$num) {
    $num = 1;
}
my $dur = $ARGV[3];
if (!$dur) {
    $dur = "10:0";
}

$u1->command("/SEQA USYN");
$u2->command("/SEQA USYN");

my $i;
for ($i = 0; $i < $num; $i++) {
    $u1->commandResponse("LOAD","test/b2b/A".$i." BOTH CRTE ");
    $u1->command("CUER","0 ".$dur);
    $u1->command("/SYNR REC");
    $u2->commandResponse("LOAD","test/b2b/B".$i." BOTH CRTE ");
    $u2->command("CUER","0 ".$dur);
    $u2->command("/SYNR REC");
}

$u2->command("/SYNC USYN");
$u2->command("UWAT");
```

RS-422 Pinouts

The 6-inch mini-DIN8 to male DB9 adapter cable (Silicon Graphics part number 018-0650-001) makes the DB9 end of the cable the same as that of an Origin200 server ttyd1/ttyd2 for RS-232 only, not for RS-422.

The RS-422 Mini-DIN8 to DB9 adapter cannot be used with Video Server Toolkit.

This appendix describes the pinouts for the Apple Macintosh cables.

For more information about pinouts, see *IRIS 4-Port Fast Ethernet Adapter With Asynchronous Serial XIO Board Installation Instructions* (part number 108-0751-001).

Macintosh Mini-DIN8 to Female DB9 Adapter

The Silicon Graphics RS-422 pinouts at the DB9 end of the Macintosh mini-DIN8 to female DB9 adapter cable is described in Table D-1.

Table D-1   Apple Macintosh Female DB9 Pinouts

Pin Number	Macintosh Female DB9
2	
3	
4	TX+
5	TX-
6	
7	
8	RX+
9	RX-

Index

Numbers

100BaseT, 4
525, 44
525 lines, 21
625, 21, 44, 62
625/50, 62

A

adapter, 213
adaptor board, 89
ADC-100 automation controller, 80
Alert, 101
Apple, xxv
audio port, 47
automate, 211

B

bandwidth, 6, 25
browser, 33

C

CDs, required, 32
Ciprico, 25

command

start, 36, 95, 96
stop, 96
command line arguments
logging, 100
compression format, 7
configuration
audio port, 47
configuring
logging, 99
signal, 52
configuring, FailSafe, 68
control, external devices, 77
Critical, 101

D

DB9, 213
Debugn, 101
decoder, 85
delay time, 63
demon, VCP-Recorder, 95
device
configuration, 42
device files, 42
devices, 5
external, 5
devices_defaults, 42
Diaquest, 77

V

vcpmaint, 33
VCP-Recorder
  demon, 95
  installation requirements, 9
  monitoring, 105
  pre-installation, 31
  running confirmation, 97
  starting, 36, 95
  stopping and starting, 95
VELA, 43
  installing, 36
Vela, 85
  configuring, 86
  hardware requirments, 85
video board, VINO, 8
VideoMedia, 83
video output sync, 59
Viewgraphics, 89
V-LAN, 83
VLAN, 43
vtr_ioconfig.conf, 41, 87
vtr.control.edit.delay, 63
vtr.control.ee.delay, 63
vtrd, 95
vtrd.conf, 60, 99
vtrlog, 95, 102, 103
vtr.media.dvb-vg.audioPidOffset, 91
vtr.media.dvb-vg.PCRPidOffset, 91
vtr.media.dvb-vg.port, 92
vtr.media.dvb-vg.videoPid, 91
vtr.media.graphics.enabled, 96
vtr.media.mpeg.pid_offset.pcr, 91
vtr.media.mpeg.pid.video, 91
vtrstart, 95, 96
vtrstat, 37, 97, 105

vtrstop, 96
vtr.time.slave_system_time, 65
VVTR, 38

W

Warning, 101

X

XFS, 20, 24
XIO, 4
XLV, 20
X Window, 96

219

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 007-3622-002.

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