

AUTODESK®
EFFECTS AND EDITING
2008

Configuration File Reference Guide for IRIX® Workstations

Autodesk®

© 2007 Autodesk, Inc. All rights reserved. Except as otherwise permitted by Autodesk, Inc., this publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

Certain materials included in this publication are reprinted with the permission of the copyright holder.

Portions of this software are copyright © 2.1.19 The FreeType Project (www.freetype.org). All rights reserved.

Portions relating to Python version 2.3.3 Copyright © 2001, 2002, 2003 Python Software Foundation; All Rights Reserved.

Portions relating to Python version 2.1.1 Copyright © 2001 Python Software Foundation; All Rights Reserved.

Portions relating to libxalan-c version 1.8.0 are copyright Apache version 2.0 Copyright 2004 The Apache Software Foundation. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

THIS PRODUCT IS LICENSED UNDER THE MPEG-4 VISUAL PATENT LICENSE PORTFOLIO LICENSE FOR THE PERSONAL AND NON-COMMERCIAL USE OF A CONSUMER FOR (i) ENCODING VIDEO IN COMPLIANCE WITH THE MPEG-4 VISUAL STANDARD ("MPEG-4 VIDEO") AND/OR (ii) DECODING MPEG-4 VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL AND NON-COMMERCIAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDED LICENSED BY MPEG LA TO PROVIDE MPEG-4 VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION INCLUDING THAT RELATING TO PROMOTIONAL, INTERNAL USES AND LICENSING MAY BE OBTAINED FROM MPEG LA, LLC. SEE [HTTP://WWW.MPEGLA.COM](http://WWW.MPEGLA.COM).

THIS PRODUCT IS LICENSED UNDER THE MPEG-2 PATENT PORTFOLIO LICENSE ANY USE OF THIS PRODUCT OTHER THAN CONSUMER PERSONAL USE IN ANY MANNER THAT COMPLIES WITH THE MPEG-2 STANDARD FOR ENCODING VIDEO INFORMATION FOR PACKAGED MEDIA IS EXPRESSLY PROHIBITED WITHOUT A LICENSE UNDER APPLICABLE PATENTS IN THE MPEG-2 PATENT PORTFOLIO, WHICH LICENSE IS AVAILABLE FROM MPEG LA, L.L.C., 250 STEELE STREET, SUITE 300, DENVER, COLORADO 80206.

Portions relating to libffmpeg Copyright © 2003-2006, Fabrice Bellard.

Portions relating to ALSA version 1.0.6 Copyright © 2004 Jaroslav Kysela, Abramo Bagnara, Takashi Iwai, and Frank van de Pol.

Powered by Automatic Duck. © 2006 Automatic Duck, Inc. All rights reserved.

Portions relating to Audiobogus Copyright © 1998-1999, Michael Pruett (michael@68k.org).

Portions relating to xxdiff Copyright © 1999-2004, Martin Blais. All Rights Reserved.

Portions relating to Audiofile 0.2.6, Open Inventor 2.1.5-9, and LibImageDL software are Copyright © 1991, 1999 Free Software Foundation, Inc.

Portions relating to Glew Copyright (c) 1991-9 Silicon Graphics, Inc. All Rights Reserved.

Portions relating to Mesa Copyright ©1999-2007 Brian Paul. All Rights Reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL BRIAN PAUL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Portions relating to OpenExr 1.2.1 Copyright (c) 2004, Industrial Light & Magic, a division of Lucasfilm Entertainment Company Ltd. Portions contributed and copyright held by others as indicated. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Industrial Light & Magic nor the names of any other contributors to this software may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions relating to Libpopt Copyright ©1998 Red Hat Software. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

Portions relating to DIRAC Time Stretch/Pitch Shift technology licensed from The DSP Dimension, <http://www.dspdimension.com> Developed and (c) 2005 Stephan M. Bernsee

Portions relating to Berkeley DB software Copyright ©1990-2002, Sleepycat Software. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Redistributions in any form must be accompanied by information on how to obtain complete source code for the DB software and any accompanying software that uses the DB software. The source code must either be included in the distribution or be available for no more than the cost of distribution plus a nominal fee, and must be freely redistributable under reasonable conditions. For an executable file, complete source code means the source code for all modules it contains. It does not include source code for modules or files that typically accompany the major components of the operating system on which the executable file runs. THIS SOFTWARE IS PROVIDED BY SLEEPYCAT SOFTWARE ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, ARE DISCLAIMED. IN NO EVENT SHALL SLEEPYCAT SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions copyright 1991-2006 Compuware Corporation.

Trademarks

The following are registered trademarks or trademarks of Autodesk, Inc., in the USA and other countries: 3DEC (design/logo), 3December, 3December.com, 3ds Max, ActiveShapes, Actrix, ADI, Alias, Alias (swirl design/logo), AliasStudio, Alias|Wavefront (design/logo), ATC, AUGI, AutoCAD, AutoCAD Learning Assistance, AutoCAD LT, AutoCAD Simulator, AutoCAD SQL Extension, AutoCAD SQL Interface, Autodesk, Autodesk Envision, Autodesk Insight, Autodesk Intent, Autodesk Inventor, Autodesk Map, Autodesk MapGuide, Autodesk Streamline, AutoLISP, AutoSnap, AutoSketch, AutoTrack, Backdraft, Built with ObjectARX (logo), Burn, Buzzsaw, CAiCE, Can You Imagine, Character Studio, Cinestream, Civil 3D, Cleaner, Cleaner Central, ClearScale, Colour Warper, Combustion, Communication Specification, Constructware, Content Explorer, Create>what's>Next> (design/logo), Dancing Baby (image), DesignCenter, Design Doctor, Designer's Toolkit, DesignKids, DesignProf, DesignServer, DesignStudio, Design|Studio (design/logo), Design Your World, Design Your World (design/logo), DWE, DWG, DWG (logo), DWG TrueConvert, DWG TrueView, DXF, EditDV, Education by Design, Extending the Design Team, FBX, Filmbox, FMDesktop, Freewheel, GDX Driver, Gmax, Heads-up Design, Heidi, HOOPS, HumanIK, i-drop, iMOUT, Incinerator, IntroDV, Inventor, Inventor LT, Kaydara, Kaydara (design/logo), LocationLogic, Lustre, Maya, Mechanical Desktop, MotionBuilder, ObjectARX, ObjectDBX, Open Reality, PolarSnap, PortfolioWall, Powered with Autodesk Technology, Productstream, ProjectPoint, Reactor, RealDWG, Real-time Roto, Render Queue, Revit, Showcase, SketchBook, StudioTools, Topobase, Toxik, Visual, Visual Bridge, Visual Construction, Visual Drainage, Visual Hydro, Visual Landscape, Visual Roads, Visual Survey, Visual Syllabus, Visual Toolbox, Visual Tugboat, Visual LISP, Voice Reality, Volo, and Wiretap.

The following are registered trademarks or trademarks of Autodesk Canada Co. in the USA and/or Canada and other countries: Backburner, Discreet, Fire, Flame, Flint, Frost, Inferno, Multi-Master Editing, River, Smoke, Sparks, Stone, Wire.

Automatic Duck and the duck logo are trademarks of Automatic Duck, Inc. All other brand names, product names or trademarks belong to their respective holders.

Disclaimer

THIS PUBLICATION AND THE INFORMATION CONTAINED HEREIN IS MADE AVAILABLE BY AUTODESK, INC. "AS IS." AUTODESK, INC., DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS.

Published by:

Autodesk, Inc.

111 McInnis Parkway

San Rafael, CA 94903, USA

Title: Autodesk Effects and Editing 2008 Configuration File Reference Guide for IRIX Workstations

Document Version: 1

Date: August 24, 2007

contents

Contents

1	Introduction	1
	Summary	1
	About Configuration Files.....	1
	About This Guide	1
	Related Documentation.....	2
	Notation Conventions	4
	Contacting Customer Support	4
2	Software Initialisation Configuration File	7
	Summary	7
	About the Software Initialisation Configuration File	7
	Default Software Initialisation Configuration File	7
	Sample Software Initialisation Configuration Files.....	8
	Viewing a Software Initialisation Configuration File	8
	Modifying a Software Initialisation Configuration File	9
	Starting with a Custom Software Initialisation Configuration File.....	11
	Updating the Software Initialisation Configuration File	11
3	Software Initialisation File Keywords	13
	Summary	13
	Introduction	14
	FrameStore.....	14
	DiskHealthNotify	15
	MemoryApplication.....	15

Video.....	17
VideoPreviewDevice.....	18
HiresChannel.....	20
TabletDriver.....	20
MidiDevice.....	21
Vtr.....	22
Emulator.....	25
Audiodevice.....	26
Gamma.....	26
3DLUT.....	27
MoncalDevice.....	28
ClipMgtDevice.....	28
MaxLibrarySize.....	31
ArchiveLibrary.....	32
SetupArchiveTape.....	32
Menu.....	33
Model.....	33
HtmlTemplates.....	33
HtmlLog.....	34
TextDefaultFont.....	34
FontDPSBase.....	34
FontDPSAFM.....	35
FontProxyLowString.....	35
FontProxyHighString.....	36
TextFileEncoding.....	36
Image File Extensions.....	37
Audio File Extensions.....	38
DefaultWebBrowser.....	38
DefaultPdfViewer.....	38
BackburnerManagerHostname.....	39
BackburnerManagerPort.....	39
BackburnerManagerPriority.....	40
BackburnerManagerGroup.....	40
CleanerDestinationPath.....	41
LogDailyReports.....	41
NetworkPanelDisplay.....	42

4 Project Configuration Files 43

Summary.....	43
About Project Configuration Files.....	43
Project Configuration File Templates.....	43

Viewing Project Configuration Files	50
Modifying Project Configuration Files	50
5 Project Configuration File Keywords	53
Summary	53
Introduction	53
Framerate	53
Timecode	54
ColourSpace.....	54
Hires	55
VideoPreviewWindow.....	56
FrameStore.....	56
ArchiveLibrary	57
Environment Directory	57
 Index	 61



Introduction

Summary

About Configuration Files	1
About This Guide	1
Related Documentation	2
Notation Conventions	4
Contacting Customer Support	4

About Configuration Files

Configuration files are ASCII text files that define the working environment for your application. Keywords in the configuration files define certain global and project-specific parameters such as I/O devices, workstation display settings, project environment directory paths, frame rate, timecode, and colour space. You can open a configuration file using any text editor, such as nedit.

There are two kinds of configuration files: the software initialisation configuration file is parsed when you launch the software, the project configuration file is parsed when you select a project in the software.

Although you should not have to regularly edit these files, it is important that you become familiar with them for the occasional instance where editing them is necessary.

About This Guide

This guide explains how to define the working environment of your Autodesk® Effects and Editing application through its software initialisation and project configuration files. Its content pertains to the following Autodesk applications for IRIX® workstations:

- Autodesk Inferno® 2008 and Autodesk Flame® 2008
- Autodesk Smoke® 2008, Autodesk Fire® 2008, and Autodesk Backdraft® Conform 2008

This guide is one of a set of three guides that you need to install and configure your Autodesk Effects and Editing hardware and software. This includes the hardware setup guide for your workstation and the software installation guide for your product.

This guide is organised as follows:

Chapter 1, “Introduction”

Learn about the information that is provided in the *Configuration File Reference Guide*, and where to find additional documentation and support, if necessary.

Chapter 2, “Software Initialisation Configuration File”

Learn how to view and modify the software initialisation configuration file to suit your needs. The software initialisation configuration file defines global software parameters, such as I/O devices, peripherals, and workstation display settings.

Chapter 3, “Software Initialisation File Keywords”

Learn about the keywords in the software initialisation configuration file that define global parameters at start-up.

Chapter 4, “Project Configuration Files”

Learn how to view and modify the project configuration files. Every project has a corresponding configuration file that defines project-specific parameters, such as frame rate, timecode, and colour space.

Chapter 5, “Project Configuration File Keywords”

Learn about the keywords in the project configuration files that define project-based parameters at start-up.

Related Documentation

The following tables list the documentation associated with the current release. For details on each of these documents, as well as help for obtaining them, refer to your application’s release notes.

User and Reference Guides	Description
<i>User’s Guide</i>	Detailed instructions on using the software
<i>What’s New</i>	A complete list of the new features for this release
<i>Online Help</i>	All of the information in the User’s Guide along with powerful search functionality

User and Reference Guides	Description
<i>Hot Keys Reference Guide</i>	A complete list of hot keys for commonly used functions
<i>Hot Keys Card</i>	A list of the most frequently used hot keys
<i>Release Notes</i>	A complete list of documentation and information on late breaking features
<i>Fixed and Known Bug List</i>	A complete list of fixed and known bugs for this release
Installation and Configuration Guides	Description
<i>Hardware Setup Guide</i> (for your workstation)	Information on how to set up your workstation's video I/O components and other peripherals
<i>Installation and Configuration Guide</i> (for your operating system)	Information on how to install and configure the Linux® or IRIX® operating system on your workstation, should you require to do so
<i>Stone and Wire Filesystem and Networking Guide</i> (for this release)	Procedures for configuring your Autodesk Stone® filesystem, Autodesk Wire® networking, and Autodesk Wiretap™ services
<i>Stone Direct Configuration Guide</i> (for this release)	Detailed connectivity diagrams and configuration procedures for you Stone storage arrays
<i>Software Installation Guide</i> (for Linux or IRIX workstations)	Information about installing and licensing your Autodesk Editing or Effects software
<i>Configuration File Reference Guide</i> (for Linux or IRIX workstations)	Information on how to modify the initialization and project configuration files associated with your Autodesk application
Other Guides	Description
<i>Using Autodesk Cleaner XL with Effects and Editing Applications</i>	Installation and troubleshooting information for Cleaner® XL
<i>Autodesk Burn Installation and User's Guide</i>	Information on how to install, set up, and use Autodesk Burn™
<i>Autodesk Backburner Installation Guide</i>	Information on how to install and set up Autodesk Backburner™
<i>Autodesk Backburner User's Guide</i>	Information on how to use Autodesk Backburner™
<i>Autodesk Wiretap Web Installation and User's Guide</i>	Information on how to install, set up, and use Autodesk Wiretap Web

Consult the Autodesk Web site at www.autodesk.com/discreet-documentation for the latest version of all documents.

Notation Conventions

A number of style conventions are used throughout this guide. These conventions and examples of their use are shown as follows.

Convention	Example
Text that you enter in a command line or shell appears in Courier bold. You must press the Enter key after each command.	rpm -qa
Variable names appear in Courier, enclosed in angle brackets. No spaces are allowed in variable names	<variable_name>
Variables that appear enclosed in square brackets are optional.	[<filename>]
Feedback from the command line or shell appears in Courier.	limit coredumpsize
Directory names, filenames, URLs, and command line utilities appear in italics.	<i>/usr/discreet</i>

Contacting Customer Support

You can contact Autodesk Media and Entertainment Customer Support at www.autodesk.com/support or through one of the following ways:

Location:	Contact Information:
Within the Americas:	Hotline (North America): 1-800-925-6442 Direct dial: 415-507-5256 (Country code = 1) Within Montreal: 514-954-7199 8 AM to 8 PM EST Monday to Friday, excluding holidays <i>me.support@autodesk.com</i>
Within Europe, Middle-East and Africa:	Hotline (from London, UK): +44 207 851 8080 9 AM to 5:30 PM (local time) Monday to Friday, excluding holidays <i>me.emea.support@autodesk.com</i>
Within Asia Pacific: (Excluding India, China, Australia, New Zealand and Japan)	Hotline (from Singapore): +65 6555 0399 9 AM to 6 PM (local time) Monday to Friday, excluding holidays <i>me.support.singapore@autodesk.com</i>
Within India:	Hotline (from Mumbai): +91 22 66952244 9:30 AM to 6:30 PM (local time) Monday to Friday, excluding holidays <i>me.support.india@autodesk.com</i>

Within Japan:	<p>Hotline (from Tokyo): 0120 107 290 Direct dial: +81 3 62211810 10 AM to 6 PM (local time) Monday to Friday, excluding holidays <i>med-sys-support-jp@autodesk.com</i></p>
Within China:	<p>Direct dial: +86 10 65056848 9 AM to 6 PM (local time) Monday to Friday, excluding holidays <i>me.support.china@autodesk.com</i></p>
Within Australia and New Zealand:	<p>Hotline (within Australia to Melbourne): +1 300 368 355 Hotline (within New Zealand to Melbourne): 0800 555301 Direct dial: +61 3 9876 8355 8 AM to 6 PM AEST Monday to Friday, excluding certain holidays <i>me.support.anz@autodesk.com</i></p>

Customer support is also available through your Autodesk reseller. To find a reseller near you, consult the reseller look-up database on the Autodesk web site at www.autodesk.com/resellers.

Software Initialisation Configuration File



Summary

About the Software Initialisation Configuration File	7
Default Software Initialisation Configuration File	7
Sample Software Initialisation Configuration Files	8
Viewing a Software Initialisation Configuration File	8
Modifying a Software Initialisation Configuration File	9
Starting with a Custom Software Initialisation Configuration File	11
Updating the Software Initialisation Configuration File	11

About the Software Initialisation Configuration File

The software initialisation configuration file (by default, *init.cfg*) is a file that is parsed when you launch the software. Keywords in the configuration file define parameters such as I/O devices, peripherals, workstation display settings, and environment directory paths. These parameters are independent of the project you select from the Project Management menu when you begin a session.

The software initialisation configuration file is an ASCII text file that you can edit using any text editor, such as nedit.

Default Software Initialisation Configuration File

When you first install the software, you are prompted to edit the software initialisation configuration file, *init.cfg*. This is the default software initialisation configuration file. It is generated from a template that corresponds to the preferred resolution that you select during the installation process.

The *init.cfg* file is located in the `/usr/discreet/<product_home>/cfg` directory.

NOTE: The `<product_home>` directory is the home directory for the current software version. For example, the home directory for Smoke is `smoke_<version>`.

As you continue to work with the software, you may have to edit the software initialisation configuration file under certain conditions, such as:

- After making changes to your hardware configuration
- When setting session-based keyword values such as specifying an archiving device

You may choose to create different versions of the software initialisation configuration file to accommodate session-based needs that change through the course of a project. To launch the software with a custom software initialisation configuration file, use the **-c** start-up option. See [“Starting with a Custom Software Initialisation Configuration File”](#) on page 11.

Sample Software Initialisation Configuration Files

Sample software initialisation configuration files are provided that define expected parameters based on the usual needs of working with NTSC, PAL, HDTV, or film material. The *init.cfg* file created for your system during the installation process is created using one of these sample files. The file used as a template depends on the selection you make when prompted to choose a preferred resolution. This selection in no way prevents you from working with other resolutions.

These samples are located in the */usr/discreet/<product_home>/cfg/<platform>/sample* directory. You should never edit the sample configuration files. Instead, either make a copy of the sample software initialization configuration file that you like or just edit the real *init.cfg* file.

The sample configuration file:	Sets parameters to:
<i>init_film.cfg</i>	Initialise the software to work with film projects.
<i>init_video.cfg</i>	Initialise the software to work with video projects.

These samples are starting points toward configuring the software to best suit your needs. Learn more about reading and modifying software initialisation configuration files in the sections:

- [“Viewing a Software Initialisation Configuration File”](#) on page 8
- [“Modifying a Software Initialisation Configuration File”](#) on page 9

Viewing a Software Initialisation Configuration File

Using a text editor, you can open a software initialisation configuration file for viewing. For example, you may want to read the keywords in different software initialisation configuration samples to see the differences. This is a good way to learn more about configuration keywords.

To open a software initialisation file for viewing:

1. Log in to the application account.

2. In a UNIX shell, type:

```
d1cfg
```

Note the following as you view the file:

- A pound sign (#) at the beginning of a line indicates a comment. These lines are not parsed during software or project initialisation.
- When you scroll through a file, underlined headings indicate distinct configuration keywords.
- Keyword headings begin with a brief description of the corresponding keyword.
- Keyword headings also include syntax rules for the corresponding keyword, with examples.
- In some cases, keyword examples are not commented out. In such cases, the keyword examples are the default settings. They are parsed during software initialisation and the corresponding parameter is set.

Modifying a Software Initialisation Configuration File

Software initialisation configuration files have open read/write permissions. This means anyone can modify them. In some cases, you may want to change the permissions associated with the files. For example, if you create a custom software initialisation configuration file to work in film, you may want to restrict access to it using the *chmod* command to prevent others from modifying it. For information on using the *chmod* command, type **man chmod** in a UNIX shell.

Here are some basic guidelines for modifying software initialisation configuration files:

- Do not modify the sample software initialisation configuration files. These are used by the application to generate new software initialisation configuration files and should not be altered.
- Create a backup of the file before making any changes to it.
- Changes to *init.cfg* may be required depending on your workstation's hardware configuration. To determine whether you have to modify the *init.cfg* file, see [“Software Initialisation File Keywords”](#) on page 13.
- When you modify a software initialisation configuration file, you must restart the software for the changes to take effect.
- You may want to make more than one software initialisation file available for use. In this case, you should make a copy of the *init.cfg* file and then modify the copy as needed. The procedure for modifying configuration files appears following this list of guidelines. To launch the

software with a custom software initialisation configuration file, use the **-c** start-up option. See [“Starting with a Custom Software Initialisation Configuration File”](#) on page 11.

To create a custom software initialisation configuration file:

1. Log in as root.
2. In a UNIX shell, go to the directory that contains the *init.cfg* file. Type:
cd /usr/discreet/<product_home>/cfg
3. Create a copy of the *init.cfg* file. Type:
cp init.cfg <custom_name>.cfg
where *<custom_name>* is the name of the new custom configuration file you will modify.
4. To open the *<custom_name>.cfg* file in a text editor, type:
nedit <custom_name>.cfg
A *nedit* window appears in which you can view and edit the *<custom_name>.cfg* file.
5. Edit the configuration file as needed. For example, you can set project archiving parameters to create archives using a VTR. To do this, scroll to the “CLIPMGTDDEVICE” keyword heading.
The description below the keyword heading indicates this keyword is used to define the devices used for archiving.
6. Scroll to the “Example for VTR” line under the “CLIPMGTDDEVICE” keyword heading.
The commented-out text below this line describes the syntax for the keyword and provides an example:

```
#ClipMgtDevice Vtr, 01:00:00:00
```
7. Place the cursor to the left of the pound sign at the beginning of the example line and press **DELETE**.
Archiving is now set to use a VTR as the archiving device. In this case, you also have to enable a VTR under the VTR keyword heading.
8. To comment out a keyword, place the cursor at the beginning of the line and press # (pound sign).
9. Press **CTRL+S** to save the *<custom_name>.cfg* file.
10. Close the *nedit* window.

For more information on configuration keywords, see [“Software Initialisation File Keywords”](#) on page 13.

Starting with a Custom Software Initialisation Configuration File

If you create a custom software initialisation configuration file, you can launch the application using a command that specifies the custom configuration file instead of the *init.cfg* file.

NOTE: When you start the software without specifying a configuration file, *init.cfg* is used. If there is no *init.cfg* file, *<application_name>.cfg* is used. If neither of these configuration files exist, the software fails to initialise.

To start the software using a custom software initialisation configuration file:

1. Log in to the application account.
2. In a UNIX shell, type:

```
<application_name> -c <custom_name>.cfg
```

where *<custom_name>* is the name of your custom software initialisation configuration file. For example, to start Smoke with a customized software initialisation configuration file called *VideoProject.cfg*, you would type:

```
smoke -c VideoProject.cfg
```

HINT: If you saved your custom software initialisation configuration file in a directory other than */usr/discreet/<product_home>/cfg*, be sure to type the full directory path.

Updating the Software Initialisation Configuration File

The *xdiff* utility is normally used during software installation to transfer facility-specific modifications in a customized software initialization configuration file from an earlier application version to the current one.

The following procedure shows how to use *xdiff* to compare old and new configuration files. While you may never need to do this, this procedure can be useful for troubleshooting problems in your facility. For example, you can use *xdiff* to troubleshoot problems in a workstation by comparing its *init.cfg* file against the *init.cfg* file of a workstation without problems.

NOTE: If you are upgrading from an earlier version of the application, several changes may have been applied to the software initialisation configuration file. Because of this, you may want to select the entire contents of the new *init.cfg* configuration file and then customize it to avoid adding obsolete keywords.

To update the software initialisation configuration file:

1. In a UNIX shell, start *xdiff* and load the two files for comparison by typing the following:

```
xdiff <old_file> <new_file>
```

For example, to launch *xdiff* to compare the previous and current software configuration files, you would type:

```
xdiff /usr/discreet/<product_home1>/cfg/init.cfg  
/usr/discreet/<product_home2>/cfg/init.cfg
```

where *<product_home1>* is the home directory for the previous release and *<product_home2>* is the home directory for the current release.

The previous and new configuration files appear beside each other in an *xdiff* window. The left panel displays the previous file settings and the right panel displays the current file settings. The path to both files is displayed in the title bar. The vertical bar on the right indicates where differences exist between the two panels.

2. Scan the old configuration file for custom directories or keywords that you want to copy to the new *init.cfg* file.

On a line-by-line basis, the differences between the old and new configuration files are highlighted.

3. For each highlighted difference, click the line that you want to keep:

- To keep the setting in the old configuration file, click the line in the left window.
- To keep the setting in the new configuration file, click the line in the right window.

Highlighted lines change colour when you click them. You must select a line in the left or right window for each difference detected by *xdiff*.

4. When you have finished updating, choose File | Save As Right to save the selected line differences under the name *init.cfg*.
5. Click OK to confirm, and then exit *xdiff*.

3

Software Initialisation File Keywords

Summary

<u>Introduction</u>	14
<u>FrameStore</u>	14
<u>DiskHealthNotify</u>	15
<u>MemoryApplication</u>	15
<u>Video</u>	17
<u>VideoPreviewDevice</u>	18
<u>HiresChannel</u>	20
<u>TabletDriver</u>	20
<u>MidiDevice</u>	21
<u>Vtr</u>	22
<u>Emulator</u>	25
<u>Audiodevice</u>	26
<u>Gamma</u>	26
<u>3DLUT</u>	27
<u>MoncalDevice</u>	28
<u>ClipMgtDevice</u>	28
<u>MaxLibrarySize</u>	31
<u>ArchiveLibrary</u>	32
<u>SetupArchiveTape</u>	32
<u>Menu</u>	33
<u>Model</u>	33
<u>HtmlTemplates</u>	33
<u>HtmlLog</u>	34
<u>TextDefaultFont</u>	34
<u>FontDPSBase</u>	34
<u>FontDPSAFM</u>	35
<u>FontProxyLowString</u>	35

<u>FontProxyHighString</u>	36
<u>TextFileEncoding</u>	36
<u>Image File Extensions</u>	37
<u>Audio File Extensions</u>	38
<u>DefaultWebBrowser</u>	38
<u>DefaultPdfViewer</u>	38
<u>BackburnerManagerHostname</u>	39
<u>BackburnerManagerPort</u>	39
<u>BackburnerManagerPriority</u>	40
<u>BackburnerManagerGroup</u>	40
<u>CleanerDestinationPath</u>	41
<u>LogDailyReports</u>	41
<u>NetworkPanelDisplay</u>	42

Introduction

This section provides a comprehensive list of the keywords that appear in a software initialisation configuration file for the application. The default software initialisation configuration file, *init.cfg*, is created from a template based on the preferred resolution you choose during the software installation process.

FrameStore

The FrameStore keyword defines the default disk storage used to store clips. It must always point to a valid, and mounted, volume of a framestore.

While you can maintain multiple framestore volumes (for example, *stonefs0* and *stonefs1*), you must choose one of those volumes to be the default. Do this by setting the FrameStore keyword to use that volume.

You can always access other available volumes (on local or remote framestores) from within the application. Refer to the chapter “Managing Projects and Users” in the application User’s Guide.

NOTE: Within the Effects or Editing application, the volume selection interface will always default to the framestore and volume selected during the previous session; it will not consider the FrameStore keyword. But, it must be defined, nonetheless, for the application to start.

The FrameStore keyword uses the following syntax:

```
FrameStore <device>
```

where <device> is the volume used to store clips.

Example of use:

```
FrameStore stonefs
```

When using Backdraft Conform, you can specify a remote framestore instead of a local one.

The FrameStore keyword for remote framestores uses the following syntax:

```
FrameStore <device>/<host name>
```

Where:	Is:
<device>	The volume used to store clips.
<host name>	The workstation to which the framestore is attached.

Example of use:

```
FrameStore stonefs/montreal1
```

NOTE: Remote access of a framestore requires a properly configured Wire network. Refer to the *Stone and Wire Filesystem and Networking Guide*.

DiskHealthNotify

The DiskHealthNotify keyword defines an e-mail address to which a message is sent if any framestore disk health problems occur. Only one address can be listed. The mailer daemon for the workstation must also be properly configured for notification e-mails to be sent.

The DiskHealthNotify keyword uses the following syntax:

```
DiskHealthNotify <e-mail_address>
```

where <e-mail_address> is the address to which you want the message sent.

Example of use:

```
DiskHealthNotify sysadmin@here.com
```

MemoryApplication

The MemoryApplication keyword allocates memory for frame buffers. A frame buffer is a chunk of memory set aside to store image data in RAM for the fastest possible display of frames while working in the application.

In most cases, the MemoryApplication keyword should be commented out. When it is commented out, the default memory configuration is used, and is based on the physical

memory installed on the system. This default configuration is sufficient for most projects, regardless of the different resolutions you may be working with.

Uncommenting the `MemoryApplication` keyword overrides the default memory configuration. You should only override the default memory configuration by uncommenting the `MemoryApplication` keyword under the following conditions:

- You are working with 4K 12-bit material.
- You are working in the application and receive “cannot allocate memory buffer” messages.
- Your system fails due to memory panic.

The `MemoryApplication` keyword uses the following syntax:

```
MemoryApplication <megabytes>
```

where `<megabytes>` is the amount of memory dedicated to the corresponding token, in megabytes.

Example of use:

```
MemoryApplication 400
```

Working with 4K 12-bit Film

Even with the recommended amount of RAM, working with 4K 12-bit film puts great demands on memory management. Set the RAM allocated for buffering frames to between 1500 and 1700.

The ability to mix different resolutions in a 4K 12-bit film project increases memory requirements even further. Even with an override of the default memory management configuration, working at this resolution may not support some advanced features that require a large amount of memory such as the Colour Warper™.

Cannot Allocate Memory Buffer Messages

“Cannot allocate memory buffer” messages indicate a shortage of memory allocated to frame buffering, so uncommenting the `MemoryApplication` keyword and defining an amount of memory for frame buffers can resolve this problem. The amount to set in this case depends on how much RAM your system has: start with 400 MB for **Editing applications** or 700 MB for **Effects applications**. Allocate more memory to frame buffers in increments of 50 MB as required.

Memory Panic

A memory panic indicates the system is running out of heap memory. Heap memory is memory allocated for use by applications and the operating system. Make sure no other applications are running. If memory panic still occurs, lower the amount of memory allocated to frame buffers.

You can monitor memory allocation process-to-process using the *gmemusage* utility. For more information, type **man gmemusage** in a UNIX shell. Remember that overriding the default memory management configuration is only required under these specific conditions. If your system experiences problems related to memory management under normal conditions, contact Customer Support.

Video

The Video keyword initialises the video device and specifies the video input type.

Video device — The video device value corresponds to the video board on your SGI®. The video input device supported for IRIX workstations is XT-DIGVID.

Default video input format — The default video input format that corresponds to the input configuration of the video board. This setting is overridden by the video input formats specified by the lines you uncomment for each VTR under the Vtr keyword heading.

The Tezro™ and the Onyx 350 accept:

- A single-link SDI input to SD Input 1 of the video breakout box (VBOB), or a single-link HD-SDI input to the HD Input 1 of the VBOB
- A dual-link SDI input to SD Input 1 and 2 of the VBOB, or a dual-link HD-SDI input to the HD Input 1 and 2 of the VBOB

The Video keyword uses the following syntax:

```
Video <video_device>, <default_video_format>
```

Where:	Is:
<video_device>	The video board used by your IRIX workstation. Its value is XT-DIGVID on Tezro and Onyx 350.
<default_video_format>	The video input type. The supported video input types are: Serial1 and SerialDual.

Examples of use:

Platform	Keyword Example
Tezro	Video XT-DIGVID, Serial1
Onyx 350	Video XT-DIGVID, Serial1

VideoPreviewDevice

The VideoPreviewDevice keyword specifies the device used for the graphics-to-video display. Specify entries for the resolutions of the projects on which you will be working as well as the resolutions supported by your hardware configuration. The *init.cfg* file contains all supported tokens for the VideoPreviewDevice keyword. The project configuration file will determine which one of these should be enabled for a given project.

VideoPreviewDevice is also used in conjunction with the VideoPreviewWindow keyword in the project configuration template files; VideoPreviewWindow specifies the width and height of the window and the refresh rate. See [“VideoPreviewWindow”](#) on page 56.

When the application is started and a project is selected, the VideoPreviewWindow must match one of the VideoPreviewDevice entries. If no match is found, graphics-to-video output is unavailable.

The VideoPreviewDevice keyword applies to:

- Tezro workstations configured with either the DVI output driving the DM5 graphics-to-video board in the VBOB, or graphics readback if DM5 hardware is not available.
- Onyx 350 workstations configured with the HDGVO graphics-to-video board. The DG5-2-TVO (HDGVO) display generator board provides a 601 and HDTV graphics-to-video path.

The VideoPreviewDevice uses the following syntax:

```
VideoPreviewDevice <description>, <device>, <channel_number>,
<xsize>, <ysize>, <refresh>, <syncsource> [, <xoffset>, <yoffset>]
```

Where:	Is:
<description>	A string describing the resolution and sync configuration of the video preview device: <i>ntsc</i> , <i>pal</i> , or an HD resolution and timing.
<device>	The graphics-to-video configuration for your system. On Tezro workstations, use <i>dm5</i> or <i>readback</i> . On Onyx 350 workstations, use <i>hdgvo</i> if you are using the HDGVO display generator. Also, for Onyx 350 workstations, you may use one of the analog display channels instead. In this case, use <i>channel</i> and specify the channel number in <channel_number>.

Where:	Is:
<channel_number>	The channel number of the connection for the specified device. On Tezro systems, use 2, the channel to which the GVO on the DM5 board is connected. Also use 2 for Onyx 350 workstations, since it is the channel to which the HDGVO board is connected. If you are using an analog channel on the Onyx 350, use the channel number, either 0 or 1.
<xsize>, <ysize>	The width and height of the area of the image window output to the graphics-to-video display.
<refresh>	The refresh rate of the combination of the HiresChannel and VideoPreviewDevice channel values.
<syncsource>	The reference sync. Set to 601sync when you are using NTSC or PAL timing and you are connected to an external sync generator. Set to DTVsync when you are using HD timing and you are connected to an external sync generator. Set to freesync when you are not connected to a sync generator.
<xoffset>, <yoffset>	The horizontal and vertical offset of the video sent to the broadcast monitor relative to the graphics display. These parameters are required for some resolutions to display accurate field dominance during playback.

Examples of use:

```
VideoPreviewDevice ntsc, dm5, 2, 720, 486, 30, 601sync
```

```
VideoPreviewDevice 1280x720@5994p, dm5, 2, 1280, 720, 5994p, DTVsync
```

NOTE: Use 1920x1080@25p_DTVsync only with DM5 and with a Tezro workstation.

```
VideoPreviewDevice ntsc, gvo, 2, 720, 486, 30, 601sync, 0, 1
```

```
VideoPreviewDevice 1280x720@5994p, channel, 1, 1280, 720, 5994p,  
601sync
```

```
VideoPreviewDevice 1920x1080@24000p, hdgvo, 2, 1920, 1080, 24000p,  
DTVsync
```

```
VideoPreviewDevice ntsc_readback, readback, 2, 720, 486, 30,  
601sync
```

When working with variable frame rate material, enable the video preview device token corresponding to the 720p timings, for example:

```
VideoPreviewDevice 1280x720@5994p, channel, 1, 1280, 720, 5994p, 601sync.
```

Configuring Readback Mode

On Onyx 350 systems without the DM5 hardware, you must enable the following lines to ensure video preview using the readback mode:

```
VideoPreviewDevice pal, readback, 2, 720, 576, 25, 601sync
```

```
VideoPreviewDevice ntsc, readback, 2, 720, 486, 30, 601sync
```

HiresChannel

The HiresChannel keyword determines the channel on which the display generator displays the user interface of the application. For some HDTV configurations, you may want to display the user interface on the second channel.

The HiresChannel keyword uses the following syntax:

```
HiresChannel <display_generator>, <channel>
```

Where:	Is:
<display_generator>	The name of the display generator used to display the user interface of the application. On Tezro, only VPRO is supported.
<channel>	The value that corresponds with the channel on the graphics board to which the graphics monitor is connected. It is usually connected to channel 0.

Examples of use:

```
HiresChannel vpro, 0
```

```
HiresChannel channel, 1
```

TabletDriver

The TabletDriver keyword identifies the tablet driver location and serial port. If you are using the Wacom® tablet driver provided by Autodesk, you must specify the serial port to which the tablet is connected.

The TabletDriver keyword uses the following syntax:

```
TabletDriver <driver_type>[, ttyd<serial_port>]
```

Where:	Is:
<driver_type>	Either dlwacom if you are using the tablet driver provided by Autodesk, or wacom if you are using another tablet driver.

Where:	Is:
<serial_port>	A parameter, used only when you use the Wacom tablet driver provided by Autodesk, to identify the serial port to which the tablet is connected.

Examples of use:

Tablet Driver	Keyword Example
Wacom tablet driver provided by Autodesk	TabletDriver dlwacom, ttyd2
Wacom tablet driver not provided by Autodesk	TabletDriver wacom

If you do not use the tablet driver provided by Autodesk, drivers for Wacom's SD and UD tablets are included in the IRIX package *oe.sw.opt.input*. The driver for the Wacom Intuos tablet must be obtained from Wacom directly—visit their Web site at www.wacom.com.

MidiDevice

The MidiDevice keyword identifies MIDI devices that are used with Autodesk Effects and Editing systems. You can uncomment multiple MIDI devices to use equipment like ADAT converters or external audio consoles at the same time.

The MidiDevice keyword uses the following syntax:

```
MidiDevice <name>, <device_configuration_file>, <serial_port>,
<protocol>, <baud_rate>, <parity>[, <stopbit>]
```

Where:	Is:
<name>	The name by which you want to identify the MIDI device in the application.
<device_configuration_file>	The name and path of the device configuration file for the MIDI device. The path is optional.
<serial_port>	The serial port to which the MIDI device is connected.
<protocol>	The protocol used to communicate with the MIDI device. It can be either <code>direct_RS422</code> or <code>direct_RS232</code> .
<baud_rate>	The baud rate used to communicate with the MIDI device.
<parity>	The parity setting to communicate with the MIDI device. It can be set to <code>EVEN</code> , <code>ODD</code> , or <code>NOPARITY</code> .
<stopbit>	The size of the stop bit (optional).

Examples of use:

MIDI Device:	Keyword Example:
Lucid ADAT Converter (MIDI Interface)	MidiDevice myMidiDevice, midi_LUCID_ADA8824, /dev/ttyd5, direct_RS422, 31250, NOPARITY
Lucid ADAT Converter (RS-232 Serial Interface)	MidiDevice myMidiDevice, midi_LUCID_ADA8824_A232, /dev/ttyd5, direct_RS232, 9600, NOPARITY
JL Cooper 3800 Audio Console (RS-422 Interface)	MidiDevice myMidiDevice, midi_JLC_MCS3800, /dev/ttyd5, direct_RS422, 38400, ODD

Vtr

The Vtr keyword identifies video tape recorders that can be used for clip I/O. You can uncomment VTRs of different video formats. Any enabled VTR can be selected for a project, regardless of the project's video I/O timings.

You can also use the Vtr keyword to identify supported High-Speed Dual-Link (HSDL) devices for clip I/O, such as telecines. HSDL devices are supported for clip I/O in Flame, Inferno, Smoke 2K, Fire, and Backdraft Conform. HSDL devices appear as separate entries in the list of supported decks for the Vtr keyword.

The Vtr keyword uses the following syntax:

```
Vtr <protocol>, <name>, <input_format>, <timing>,
<colorspace_mode>, <output_format>, <output_sync>, <serial_port>,
<timecode_type>, <video_output_delay>, <video_input_delay>,
<pre_roll>, <post_roll>, <audio_input_delay>,
<audio_output_delay>, <video_precision>
[, <cueup_mode>, <TC_transition_delay>, <edit_on_delay>,
<edit_off_delay>, <vtr_command_delay>]
```

Where:	Is:
<protocol>	The VTR control protocol (SONY, BTS, BVW50, TASCAM, or NONE).
<name>	The name by which you want to identify the VTR in the Input Clip and Output Clip menus (D1, DigBeta, D1 BTS, D5, and DVCpro, for example.).
<input_format>	The video input format. Set to Serial1 to input using a single-link (4:2:2) connection from a device. Set to SerialDual to input using a dual-link (4:4:4) connection from a device.

Where:	Is:
<timing>	The I/O timing associated with the video standard of the VTR (NTSC, PAL, or HD, if applicable).
<colorspace_mode>	The mode that indicates whether colourspace conversion and/or headroom is required for clip I/O.
<output_format>	The video output type. Set to <code>Serial1</code> to output using a single-link (4:2:2) connection to a device. Set to <code>SerialDual</code> to output using a dual-link (4:4:4) connection to a device.
<output_sync>	The sync source used for clip output.
<serial_port>	The serial port to which the VTR is connected. See “Serial Port Definition for VTRs” on page 24.
<timecode_type>	The timecode type to be returned by the VTR (Auto, LTC, or VITC).
<video_output_delay>	Video output delay in frames.
<video_input_delay>	Video input delay in frames.
<pre_roll>	Pre-roll in frames or seconds: Use integers to specify pre-roll in frames. Use decimals to specify pre-roll in seconds.
<post_roll>	Post-roll in frames or seconds: Use integers to specify post-roll in frames. Use decimals to specify post-roll in seconds.
<audio_input_delay>	The offset value to have sync audio with video on input. Integer units represent frames; this delay should be 0 by default.
<audio_output_delay>	The offset value to have sync audio with video on output. Integer units represent frames; this delay should be 0 by default.
<video_precision>	The precision of the video interface (8 or 10 bits).
<cueup_mode>	The method by which the VTR is cued. This parameter is optional. Use <code>vtrcueing</code> to make the application use the VTR Cue command directly to cue the VTR. Use <code>vtrfff</code> if using <code>vtrcueing</code> causes the VTR to react slowly when cued, such as with a Betacam SP™.
<TC_transition_delay>	The delay in milliseconds after a vertical sync, before requesting the VTR timecode. This value should only be set or changed with the help of technical support.
<edit_on_delay>	The delay in frames before the edit sync point to send the ON command. This parameter is optional and it applies only to BVW50.

Where:	Is:
<code><edit_off_delay></code>	The delay in frames before the edit sync point to send the OFF command. This parameter is optional and it applies only to BVW50.
<code><vtr_command_delay></code>	The delay in milliseconds before the application sends certain commands to the VTR. This parameter is optional, but useful for older VTRs such as the BTS.

NOTE: The *init.cfg* file contains a comprehensive list of VTR keyword examples for supported VTRs and HSDL devices. In most cases you just have to uncomment the lines for the device that you want to use. In some cases minor changes to parameters are required.

Serial Port Definition for VTRs

The Vtr keyword examples in the software initialisation configuration file are set to support VTRs connected to serial ports in a typical configuration. A VTR is usually connected to serial port 1, in which case its `<serial_port>` definition should be:

```
/dev/tty4d1
```

On Onyx 350 workstations, VTRs connect to serial port 5 (*/dev/tty4d5*). If you connect a VTR to serial port 1, you must disable console terminal/modem use for that serial port. This is done automatically if you choose to make serial port 1 available to the application during the software installation process. Otherwise, you must disable console terminal/modem use manually.

To disable the console manually:

1. Log in as root.
2. Open the */etc/inittab* file in a text editor. In a UNIX shell, type:

```
nedit /etc/inittab
```

3. Search for lines beginning with `t1:23`.

The search should bring you to a line that looks like this:

```
t1:23:respawn:/sbin/suattr[...]
```

4. If this line says “respawn”, change it to:

```
t1:23:off:/sbin/suattr[...]
```

5. Save and close the file.
6. Enable the modification. Type:

```
telinit q
```

Configuring Colourspace Conversion for Device I/O

When you set the input or output format for a VTR or HSDL device in the `Vtr` keyword, you should also specify its colourspace conversion method using the `<colourspace mode>` parameter. The supported conversion methods depend on whether a single-link or dual-link connection is used for I/O with the device. The following table shows the colourspace conversion methods that are available for single-link and for dual-link I/O connections on Linux workstations.

Colourspace Conversion Method	Supported for Single-Link Serial (4:2:2) I/O	Supported for Dual-Link Serial (4:4:4) I/O
YCbCR -> RGB	Yes	No
YCbCR -> RGB + Headroom	Yes	No
No Conversion	No	Yes
No Conversion + Headroom	No	Yes

Emulator

Use the `Emulator` keyword to enable the VTR Emulation feature and configure your workstation to emulate a Sony™ VTR that is controllable via the RS-422 serial port. For help configuring the hardware for VTR emulation, see the *Hardware Setup Guide* for your workstation.

The VTR Emulation feature supports SD and HD video timings. This feature is available in Backdraft Conform, Smoke 2K, and Fire.

You can specify more than one emulator. Any enabled emulator can be selected for a project, regardless of the project's video I/O timings.

The Emulator keyword uses the following syntax:

```
Emulator sony, <name>, <input_format>, <timing>,
<colourspace_mode>, <output_format>, <output_sync>, <serial_port>,
<emulator_output_delay>, <emulator_input_delay>,
<audio_input_delay>, <audio_output_delay>, <video_precision>
```

Where:	Is:
<code><name></code>	The name for the emulator.
<code><input_format></code>	The video input format. For a list of the video formats supported by your workstation, see "Video" on page 17.
<code><timing></code>	The video resolution and timing of the VTR to be emulated.
<code><colourspace_mode></code>	The colourspace mode for video transfers.

Where:	Is:
<output_format>	The video output format. For a list of the video formats supported by your workstation, see "Video" on page 17.
<output_sync>	The source used to synchronize the video output. By default, this value is set to <i>Autodetect</i> .
<serial_port>	The serial port to which the VTR control cable is connected.
<emulator_output_delay>	The video output delay used by the emulator in frames.
<emulator_input_delay>	The video input delay used by the emulator in frames.
<audio_input_delay>	The audio input delay used by the emulator in frames.
<audio_output_delay>	The audio output delay used by the emulator in frames.
<video_precision>	The interface precision used for video transfers.

Examples of use:

```
Emulator sony, NTSC, SERIAL1, NTSC, YCBCR_RGB_CONVERSION,
SERIAL1, AUTODETECT, /dev/tty4d3, -4, 0, 0.00, 0.00, 8
```

```
Emulator sony, HSDL 1499 sf, SERIALDUAL, 2048x1556_1499SF,
NO_CONVERSION, SERIALDUAL, STANDALONE, /dev/tty4d3, -4, 0, 0.00,
0.00, 8
```

Audiodevice

The Audiodevice keyword initialises the specified audio device.

The Audiodevice keyword uses the following syntax:

```
AudioDevice <type>
```

where <type> is *sgi*.

sgi specifies the Discreet® Native Audio subsystem. In Editing applications, the Discreet Native Audio subsystem supports up to eight audio I/O channels. In Effects applications, Discreet Native Audio supports two channels of audio I/O.

Example of use:

```
Audiodevice sgi
```

Gamma

The Gamma keyword sets the gamma correction value for the graphics monitor. It only affects the display of the images on the graphics monitor; it does not affect image processing or video I/O.

More than one gamma setting can be specified. If you specify multiple gamma settings:

- The first setting is used by the application at start-up.
- Subsequent settings are stored in the Preferences menu. You can select another setting to change the gamma correction setting during a work session.

For information about using gamma correction to adjust how images appear on a graphics monitor, refer to the chapter “Monitor Calibration” in the User’s Guide of your Effects or Editing application.

The Gamma keyword uses the following syntax:

Gamma <value>

where <value> is either a floating point value or a filename. Use the following information to correctly format its value.

- If you use a floating point value (for example, 1.0), it is used as the gamma correction setting.
- If you specify a filename, it must end with either a .lut or .monitor extension. The gamma correction setting specified by the corresponding LUT (Look-Up Table) or monitor calibration setup is used. LUTs and monitor calibration setups must reside in their respective folders, /usr/discreet/<product_home>/lut or /usr/discreet/<product_home>/monitor. Both are in ASCII format.

Examples of use:

Gamma Definition	Keyword Example
Floating-point value	Gamma 1.0
LUT (look-up table)	Gamma custom.lut
Monitor calibration setup	Gamma calib.monitor

3DLUT

The 3DLUT keyword sets the 3D LUTs available for image preview. It has no effect on the image itself (it does not affect image processing or video I/O). 3D LUTs are supported in Flame, Inferno, Smoke 2K, Fire, and Backdraft Conform.

Encrypted (.e3d) and unencrypted (.3dl) 3D LUTs are supported. More than one 3D LUT can be specified for the keyword. If you specify multiple LUTs, they appear in the Preferences menu in the order listed in the configuration file.

The 3DLUT keyword uses the following syntax:

```
3DLUT <file_name>
```

where <file_name> is a 3D LUT stored in the directory */usr/discreet/<product_home>/lut*; otherwise, you must include the full path to the file.

Example of use:

```
3DLUT 3Dlut1.3dl
```

MoncalDevice

The MoncalDevice keyword specifies the colourimeter used for monitor calibration as well as the serial port to which it is connected. For a list of certified colourimeters, refer to the Documentation Library on the Autodesk Web site at <http://www.autodesk.com/discreet-documentation>.

The MoncalDevice keyword uses the following syntax:

```
MoncalDevice <device_name>, /dev/tty<serial_port>
```

Where:	Is:
<device_name>	The type of monitor calibration device you are using.
<serial_port>	The number of the serial port to which the monitor calibration device is attached.

Examples of use:

Colourimeter	Keyword Example
Barco Calibrator	MoncalDevice Barco Calibrator, /dev/ttyd2
X-Rite	MoncalDevice X-Rite DTP92, /dev/ttyd2

ClipMgtDevice

The ClipMgtDevice keyword defines the devices used for archiving. The following archiving devices are supported:

- VTR
- HDCAM
- Tape
- File
- DST

Using the guidelines for setting ClipMgtDevice keywords in this section, you can set multiple archiving devices for use with the application. The first device that is defined will be the default device.

ClipMgtDevice VTR

You can specify only one VTR for a VTR clip management device. The ClipMgtDevice VTR keyword contains an optional start timecode parameter:

- If you specify a default start timecode for VTR archiving, the specified timecode is always used. You cannot override the default start timecode using the Archive menu.
- If you want to determine through the application the start timecode for archives created with a VTR, do not add the start timecode parameter.

The ClipMgtDevice VTR keyword uses the following syntax:

```
ClipMgtDevice Vtr[, <timecode>]
```

where <timecode> is an optional start timecode for the archive.

Example of use:

```
ClipMgtDevice Vtr, 01:00:00:00
```

ClipMgtDevice HDCAM

To archive to a Sony HDCAM VTR, you must use the ClipMgtDevice HDCAM keyword and not the ClipMgtDevice VTR keyword. The ClipMgtDevice HDCAM keyword adjusts the metadata encoding mechanism to account for the compression method used by the HDCAM VTR.

The ClipMgtDevice HDCAM keyword contains an optional start timecode parameter:

- If you specify a default start timecode for HDCAM VTR archiving, the specified timecode is always used. You cannot override the default start timecode using the Archive menu.
- If you want to determine through the application the start timecode for archives created with an HDCAM VTR, do not add the start timecode parameter.

The ClipMgtDevice HDCAM keyword uses the following syntax:

```
ClipMgtDevice HDCAM[, <timecode>]
```

where <timecode> is an optional start timecode for the archive.

Example of use:

```
ClipMgtDevice HDCAM, 01:00:00:00
```

ClipMgtDevice Tape

You can specify several tape devices for clip management. To use a tape device for archiving, you must define the filename, the block size, and the name for the device you are using.

The ClipMgtDevice Tape keyword uses the following syntax:

```
ClipMgtDevice Tape, <file_name>, [<block_size>], [<device_name>]
```

Where:	Is:
<file_name>	The filename of the fixed block size device.
<block_size>	The amount of data per block written to tape.
<device_name>	The name of the tape device as it will appear in the Archive menu.

Examples of use:

Tape Device	Keyword Example
DAT	ClipMgtDevice Tape, /dev/rmt/tps*d*ns, 65536, DAT
Exabyte	ClipMgtDevice Tape, /dev/rmt/tps*d*ns.8500, 65536, Exabyte
DTF2	ClipMgtDevice Tape, /dev/rmt/tps*d*ns, 262144, DTF2
AIT	ClipMgtDevice Tape, /dev/rmt/tps*d*ns, 262144, AIT

NOTE: Use the `hinv -c tape` command to determine the SCSI ID and controller number for the tape drive(s) you are using.

ClipMgtDevice File

You can define a portion of your system disk or another volume as the destination for archives created using the application.

The ClipMgtDevice File keyword uses the following syntax:

```
ClipMgtDevice File, <file_name>, <size>
```

Where:	Is:
<file_name>	The path for archives created using the application.
<size>	The maximum size for a file archive in MB.

You can also modify the parameters of this keyword through the application.

- You can use a path other than the default filename by entering one in the Archive menu.

- You can change the default maximum size of a file archive in the Archive menu in the application.

Examples of use:

File Type	Keyword Example
System disk directory	ClipMgtDevice File, /usr/tmp/archive, 1024
CD-ROM	ClipMgtDevice File, /CDROM

ClipMgtDevice DST

To use a DST device for archiving, you must define the destination directory for archives created using the application.

The ClipMgtDevice DST keyword uses the following syntax:

```
ClipMgtDevice DST, <path_name>
```

where <path_name> is the path to the DST device.

Example of use:

```
ClipMgtDevice DST, /dev/rmt/tps1d4
```

MaxLibrarySize

The MaxLibrarySize keyword indicates the maximum size for any single clip library, in megabytes. The higher the value, the more memory the application uses. Using a larger value reduces memory fragmentation, which optimizes memory use. However, the value should not be so high as to compromise system performance.

The software also uses this keyword to determine whether there is enough free hard drive space available at start-up. To start the software, you must have at least 10 megabytes of free hard drive space in addition to the value set by this keyword.

The size of your libraries can be determined using the following command in a UNIX shell:

```
ls -lh /usr/discreet/clip/**/*.*.000.clib
```

The MaxLibrarySize keyword uses the following syntax:

```
MaxLibrarySize <size>
```

where <size> is the maximum library size in megabytes.

Example of use:

```
MaxLibrarySize 30
```

ArchiveLibrary

The ArchiveLibrary keyword identifies the directory to which online HTML and ASCII tables of contents are saved when archiving. Its default value is *usr/discreet/archive*. You can make the target directory relative to the home directory of the application by prefixing the path with a tilde (e.g. *~/archive*).

NOTE: The ArchiveLibrary keyword of the project configuration file, if set, has precedence over the one in the software initialisation configuration file.

The ArchiveLibrary keyword uses the following syntax:

```
ArchiveLibrary <directory_path>
```

where *<directory_path>* is the path to which online HTML and ASCII tables of contents are saved.

Example of use:

```
ArchiveLibrary /usr/discreet/archive
```

SetupArchiveTape

The SetupArchiveTape keyword identifies the device to which setup information for an archived project is saved. A project's setups are saved as a *.tar* format archive, creating a single file that can be extracted, preserving the original directory structure.

The SetupArchiveTape keyword uses the following syntax:

```
SetupArchiveTape <device_type>
```

where *<device_type>* is the path to the device where you want to save the setup archive. You can set the path to point to a tape device or use a file destination as a virtual device. If you set a file destination, you must add a filename ending with the *.tar* extension that you want to use for the setup archive.

Examples of use:

Tape Device	Keyword Example
DAT	SetupArchiveTape /dev/rmt/tps0d2ns
Exabyte	SetupArchiveTape /dev/rmt/tps0d3ns.8500
File	SetupArchiveTape /usr/tmp/setups.tar

NOTE: Enable only one setup archive device. If you uncomment more than one device, the last device overrides previous uncommented SetupArchiveTape lines.

Menu

The Menu keyword specifies where application menu files are stored. The Menu keyword is one of the environment directory keywords which specify the paths to directories for resources shared by all projects. These directory paths should not be modified.

Example of use:

```
Menu ~/menu, menu
```

where the `~` in the directory path stands for `/usr/discreet/<product_home>`.

Model

The Model keyword specifies where 3D models used by Action or DVE are stored. The Model keyword is one of the environment directory keywords which specify the paths to directories for resources shared by all projects. These directory paths should not be modified.

Example of use:

```
Model ~/dve
```

where the `~` in the directory path stands for `/usr/discreet/<product_home>`.

HtmlTemplates

The HtmlTemplates keyword specifies where HTML templates used for archive online tables of contents are stored. The HtmlTemplates keyword is one of the environment directory keywords which specify the paths to directories for resources shared by all projects. These directory paths should not be modified.

Example of use:

```
HtmlTemplates ~/templates
```

where the `~` in the directory path stands for `/usr/discreet/<product_home>`.

HtmlLog

The `HtmlLog` keyword allows you to specify a directory in which to write the Batch module HTML status and log. This keyword is used by `Inferno` and `Flameonly`.

The `HtmlLog` keyword uses the following syntax:

```
HtmlLog <directory_path>
```

where `<directory_path>` is the destination directory.

Example of use:

```
HtmlLog /usr/discreet/html
```

TextDefaultFont

The `TextDefaultFont` keyword sets the default font for the `Text`, `Paint`, and `Action` modules.

The `TextDefaultFont` keyword uses the following syntax:

```
TextDefaultFont <font>
```

where `` is the name of the font you want to set as the default.

Example of use:

```
TextDefaultFont Discreet
```

FontDPSBase

The `FontDPSBase` keyword identifies the directory in which PostScript® fonts are stored. At initialisation, the application creates links in the `/usr/discreet/font` directory that point to fonts in the directory identified by this keyword.

The `FontDPSBase` keyword uses the following syntax:

```
FontDPSBase <directory_path>
```

where `<directory_path>` identifies the directory in which fonts are stored.

Example of use:

```
FontDPSBase /usr/lib/X11/fonts/Type1
```

If you do not specify the directory pathname, the application uses */usr/lib/DPS/outline/base*. In most cases this directory—created when you installed the Display PostScript software as part of the installation—should be the one identified by the `FontDPSBase` keyword. Using this directory provides access to PostScript fonts.

NOTE: To use TrueType fonts with the application, copy these fonts into the */usr/discreet/font* directory. For the application to recognize TrueType fonts in this directory, they must have the *.ttf* extension.

FontDPSAFM

The `FontDPSAFM` keyword identifies the directory in which font metrics are stored. Font metrics provide information about each font that improves kerning. At initialisation, the application creates links in the directory */usr/discreet/font* that point to font metric files in the directory identified by this keyword.

The `FontDPSAFM` keyword uses the following syntax:

```
FontDPSAFM <directory_path>
```

where `<directory_path>` identifies the directory in which font metrics are stored.

Example of use:

```
FontDPSAFM /usr/lib/X11/fonts/Type1
```

If you do not specify a directory pathname, the application uses */usr/lib/DPS/AFM*. In most cases this directory—created when you installed the Display PostScript software as part of the installation—should be the one identified by the `FontDPSAFM` keyword.

FontProxyLowString

The `FontProxyLowString` keyword, along with the `FontProxyHighString` keyword, specifies which characters to draw in font proxies. By default, the proxy string is “Aa”. You can override the default string for non-extended character sets by uncommenting and editing the `FontProxyLowString` keyword option.

Both `FontProxyLowString` and `FontProxyHighString` keyword options can be uncommented at the same time. Extended character sets try the `FontProxyHighString` keyword first. If the values in the `FontProxyHighString` keyword option do not apply to the font, the `FontProxyLowString` keyword option is used instead.

The FontProxyLowString keyword uses the following syntax:

```
FontProxyLowString <code>[, ...]
```

Where:	Is:
code	The Unicode value associated with the character that you want to display.
...	Up to four (for a total of five) more Unicode values for the font proxy string.

Example of use:

```
FontProxyLowString 0x41, 0x61
```

This example displays the default proxy string, “Aa”.

FontProxyHighString

The FontProxyHighString keyword, along with the FontProxyLowString keyword, specifies which characters to draw in font proxies. By default, the proxy string is “Aa”. If a font includes glyph definitions for extended character sets (such as Asian character sets), you can set a proxy string by uncommenting and editing the FontProxyHighString keyword.

Both FontProxyHighString and FontProxyLowString keyword options can be uncommented at the same time. Extended character sets try the FontProxyHighString keyword first. If the values in the FontProxyHighString keyword option do not apply to the font, the FontProxyLowString keyword option is used instead.

The FontProxyHighString keyword uses the following syntax:

```
FontProxyHighString <code>[, ...]
```

Where:	Is:
code	The Unicode value associated with the character that you want to display.
...	Up to four (for a total of five) more Unicode values for the font proxy string.

Example of use:

```
FontProxyHighString 0x3042, 0x30a2
```

This example displays the Japanese “Hiragana A” and “Katakana A” characters.

TextFileEncoding

The TextFileEncoding keyword is the list of file encodings that will be supported for importing text files. The encoding must be supported for “iconv”. To get the list of supported encodings, type **iconv -l** in a UNIX shell.

NOTE: The current local encoding and UCS-2 unicode encoding are always included by default.

The TextFileEncoding keyword uses the following syntax:

```
TextFileEncoding <character_set>
```

Examples of use:

```
TextFileEncoding ISO8859-1
```

```
TextFileEncoding EUCJP
```

Image File Extensions

This list defines the supported image and movie file formats that the application can input or output. These extensions are used to filter files of the corresponding format when you use the file browser. This list may be edited to suit your particular needs.

Image Format	Extension
Alias®	als
Cineon®	cin
Digital Picture Exchange	dpx
Jpeg	jpg
Pict	pict
Pixar	picio
Sgi®	sgi
Softimage®	pic
Targa®	tga
Maya®	iff
Tiff	tif
Wavefront™	rla
Photoshop®	psd
OpenEXR	exr
Quicktime®	mov
SGI Movie	mv
MPEG-1 Video	mpg
AVI Movie	avi

Audio File Extensions

This list defines the supported audio file formats that the application can input or output. The extensions are used to filter files of the corresponding format when you use the file browser. This list may be edited to suit your particular needs.

Audio Format	Extension
AIFF (standard)	aiff
AIFFC (extended)	aifc
Sun	au
Microsoft®	wav
Berkeley (BSD)	bsf
AVR (Audio Visual Research)	avr
MPEG-1 Audio	mp1
Sound Designer II	sd2
Creative Labs VOC	voc
Sample Vision	svf
Creative Labs SoundFonts	sf2
MPEG-1, Layer 3	mp3

DefaultWebBrowser

The DefaultWebBrowser keyword identifies the Web browser used by the application to browse online help and to view HTML tables of contents for archives.

The DefaultWebBrowser keyword uses the following syntax:

```
DefaultWebBrowser <browser>
```

where <browser> identifies the Web browser you want to use.

Example of use:

```
DefaultWebBrowser mozilla
```

DefaultPdfViewer

The DefaultPDFViewer keyword identifies the PDF viewer on your workstation used to view PDFs.

The DefaultPDFViewer keyword uses the following syntax:

```
DefaultPDFViewer <PDF_viewer>
```

where <PDF_viewer> is the application you want to use to view PDFs, such as Xpdf.

Example of use:

```
DefaultPDFViewer xpdf
```

BackburnerManagerHostname

The BackburnerManagerHostname keyword serves two functions:

- It locates the **Backburner** Manager on your rendering network. Enable this keyword if you are running an Autodesk Backburner Distributed Queuing System in your facility or will be using **Cleaner XL** to encode jobs exported from the application. All four BackburnerManager keywords must be set correctly for jobs to be sent to the rendering network.
- It enables the Background Wire and Background Proxies buttons in the application; they are otherwise greyed out. Uncomment both the BackburnerManagerHostName and BackburnerManagerPort keywords to enable the buttons.

The BackburnerManagerHostname keyword uses the following syntax:

```
BackburnerManagerHostname <host_name>
```

where <host_name> is the hostname of the Windows® workstation that is running **Backburner** Manager.

Example of use:

```
BackburnerManagerHostname titan.saturn.com
```

BackburnerManagerPort

The BackburnerManagerPort keyword serves two functions:

- It sets the port where the **Backburner** Manager accepts jobs from the workstation. Enable this keyword if you are running an Autodesk Backburner Distributed Queuing System in your facility or will be using **Cleaner XL** to encode jobs exported from the application. All four BackburnerManager keywords must be set correctly for jobs to be sent to the rendering network.
- It enables the Background Wire and Background Proxies buttons in the application; they are otherwise greyed out. Uncomment both the BackburnerManagerHostName and BackburnerManagerPort keywords to enable the buttons.

The BackburnerManagerPort keyword uses the following syntax:

```
BackburnerManagerPort <port_number>
```

By default, <port_number> is set to 3234, which is the default port where the **Backburner** Manager accepts jobs from workstations on the network. Do not change this value unless you are sure that the **Backburner** Manager on your network is using a different port number.

Example of use:

```
BackburnerManagerPort 3234
```

BackburnerManagerPriority

The **BackburnerManagerPriority** keyword sets the priority for jobs created on your application for **Backburner** Manager on your rendering network. Enable this keyword if you are running an Autodesk Backburner Distributed Queueing System in your facility or will be using **Cleaner XL** to encode jobs exported from the application. All four **BackburnerManager** keywords must be set correctly for jobs to be sent to the rendering network.

The BackburnerManagerPriority keyword uses the following syntax:

```
BackburnerManagerPriority <priority>
```

where <priority> is a value from 0 to 100, where 0 is highest priority and 100 is lowest. The default is 50.

Example of use:

```
BackburnerManagerPriority 25
```

BackburnerManagerGroup

The **BackburnerManagerGroup** keyword defines the group of machines to which jobs created in your software application will be submitted. Enable this keyword if you are running an Autodesk Backburner Distributed Queueing System in your facility or will be using **Cleaner XL** to encode jobs exported from the application. All four **BackburnerManager** keywords must be set correctly for jobs to be sent to the rendering network.

The BackburnerManagerGroup keyword uses the following syntax:

```
BackburnerManagerGroup <group_name>
```

where <group_name> is the name of a group of computers on an Autodesk Burn rendering network.

Example of use:

```
BackburnerManagerGroup renderfarm1
```

CleanerDestinationPath

The CleanerDestinationPath keyword sets the default path on a Windows workstation where clips are saved after encoding by **Cleaner XL**.

The default path you enter appears in the Cleaner Destination Path field when you select Cleaner in the Format Box of the Image Export menu. You can edit the path there. If there is an ftp path in the output profile, the ftp path is used as an additional destination for encoded jobs. See “Editing and Installing Output Profiles” in the chapter “Network Encoding with Cleaner XL” of the application User’s Guide.

The CleanerDestinationPath keyword uses the following syntax:

```
CleanerDestinationPath <path>
```

By default, <path> is set to *C:\Tmp*, which is the default file path where **Cleaner XL** saves exported clips after encoding.

NOTE: Make sure the folder to which the path points exists and is accessible from the Windows workstation where Cleaner XL jobs will be encoded. Also make sure that this folder has write privileges and has enough space to accommodate encoded jobs.

Example of use:

```
CleanerDestinationPath C:\Cleaner\exported
```

LogDailyReports

The LogDailyReports keyword specifies the number of application log files that are kept. When the number of application log files on the workstation exceeds this value, the oldest log is deleted to preserve space.

The LogDailyReports keyword uses the following syntax:

```
LogDailyReports <number>
```

where <number> is the number of log files to be kept on the workstation. Set this value to 0 to keep all application log files.

Example of use:

```
LogDailyReports 10
```

NetworkPanelDisplay

The NetworkPanelDisplay keyword filters the list of framestores in the Network panel to show framestores that are either available on the network and mounted, or available on the network but with undetermined mount status.

The NetworkPanelDisplay keyword uses the following syntax:

```
NetworkPanelDisplay <filter_setting>
```

where <filter_setting> is set to either ShowMounted or ShowAll.

Use:	To:
ShowMounted	Show only framestores verified as both mounted and available on the Wire network.
ShowAll	Show all framestores available on the network without first checking whether they are mounted. Using this option slightly reduces the time needed for your application to start because no further checks are performed on remote framestores.

Example of use:

```
NetworkPanelDisplay ShowMounted
```

Project Configuration Files



Summary

About Project Configuration Files	43
Project Configuration File Templates	43
Viewing Project Configuration Files	50
Modifying Project Configuration Files	50

About Project Configuration Files

The project configuration file is a file that is parsed when you select a project in the application. A project configuration file uses the same name as the project, but with a *.cfg* extension. For example, the project configuration file for a project called “my_project” uses a project configuration file called *my_project.cfg*.

Project configuration files are located in the */usr/discreet/project/<project_name>/cfg* directory.

Project configuration files are ASCII text files that you can edit using any text editor, such as *nedit*. As a general rule, you do not have to modify this file. Learn more about reading and modifying project configuration files in the sections:

- [“Viewing Project Configuration Files”](#) on page 50
- [“Modifying Project Configuration Files”](#) on page 50

Project Configuration File Templates

When you create a project, you select a template from which your project’s configuration file is generated. These templates are based on a resolution and frame rate, and define the display environment. Project templates do not prevent you from working with clips of other resolutions and frame rates. However, some functions, such as display on a broadcast monitor, may not be supported.

These templates are located in the `/usr/discreet/<product_home>/cfg/<platform>/template` directory.

HINT: This directory contains an extensive list of templates for all supported display environments, some of which may not be suitable for your workflow. To make template selection easier, use the `mkdir` command to create a directory called `unused` in the directory `/usr/discreet/<product_home>/cfg/<platform>/template`. Use the `mv` command to move templates you do not use to this folder. When you use the Create Project menu, only those templates that remain will be available.

NOTE: You can add custom templates to the list of available ones. Refer to [“Modifying Project Configuration Files”](#) on page 50.

Tezro with the DM5 board

On a Tezro with the DM5 board, choose the template from the following table that corresponds to the default resolution that you want to work with.

The configuration file template:	Sets parameters for projects to work with:
<code>ntsc.cfg</code>	NTSC 601 material. The external sync source is NTSC blackburst.
<code>pal.cfg</code>	PAL 601 material. The external sync source is PAL blackburst.
<code>24p.cfg</code>	24p material of any resolution.
<code>24p_ntsc.cfg</code>	Film-based material that has been transferred to NTSC 601 video. The external sync source is NTSC blackburst.
<code>24p_1920x1080@5994i.cfg</code>	Film-based material that has been transferred to 1920x1080 59.94i HD video. The external sync source is NTSC blackburst.
<code>1280x720@5994p.cfg</code> <code>1280x720@6000p.cfg</code> <code>1920x1035@5994i_1080.cfg</code> <code>1920x1080@23976psf.cfg</code> <code>1920x1080@24000psf.cfg</code> <code>1920x1080@25000psf.cfg</code> <code>1920x1080@50i.cfg</code> <code>1920x1080@5994i.cfg</code> <code>1920x1080@6000i.cfg</code>	The corresponding HD standard on systems configured with the DM5 board. The external sync source corresponds with the HD format.
<code>film.cfg</code>	Film-based material.
<code>film_48.cfg</code>	Film-based material that is viewed on a widescreen LCD monitor. If <code>film.cfg</code> is used instead, the monitor will be unable to display the signal.

The configuration file template:	Sets parameters for projects to work with:
<i>film_ntsc.cfg</i>	Film-based material that has been transferred to NTSC video. The external sync source is NTSC blackburst.
<i>film_ntsc_48.cfg</i>	Film-based material that has been transferred to NTSC video and is viewed on a widescreen LCD monitor. The external sync source is NTSC blackburst. If <i>film_ntsc.cfg</i> is used instead, the monitor will be unable to display the signal.

NOTE: The *film_48.cfg* and *film_ntsc_48.cfg* templates set the refresh rate of the monitor to 48 Hz. Using these templates with a CRT monitor will result in a display that flickers noticeably. With CRT monitors, use *film.cfg* and *film_ntsc.cfg* instead.

For information on the HD formats that correspond to these project templates, see [“Supported HD Formats in Project Templates”](#) on page 49.

Onyx 350 with the DG5-2-GVO Board (Graphics Monitor on Channel 0)

On an Onyx 350 with the DG5-2-GVO board and graphics monitor on channel 0, the following devices are connected to the DG5-2-GVO board:

- A graphics monitor is connected to RGB analog channel 0.
- An HD analog RGB broadcast monitor is connected to RGB analog channel 1.
- A 601 SDI broadcast monitor is connected to the 601A link.

Sync is restricted to NTSC and PAL. You can use this sync source to genlock HD projects at 59.94 Hz (NTSC) and 50 Hz (PAL). However, the analog RGB output sent through channel 1 to the broadcast monitor is not completely stable. On some HD broadcast monitors, horizontal jitter in the upper part of the frame occurs.

You can use **_free.cfg* templates to work on HD material incompatible with NTSC and PAL genlocking, and no jitter will occur on the broadcast monitor. The disadvantage is that audio will drift with respect to video.

Image jitter and audio drift occur only on playback through the software. Video I/O is unaffected. Still, this configuration is recommended only for systems where either no HD work is done, or infrequent HD work is done.

Use the template from the following table that corresponds to the resolution and sync configuration that you want to work with.

The configuration file template:	Sets parameters for projects to work with:
<i>ntsc.cfg</i>	NTSC 601 material. The external sync source is NTSC blackburst.
<i>pal.cfg</i>	PAL 601 material. The external sync source is PAL blackburst.
<i>24p.cfg</i>	24p material of any resolution.
<i>24p_ntsc.cfg</i>	Film-based material that has been transferred to NTSC 601 video. The external sync source is NTSC blackburst.
<i>24p_1920x1080@5994i.cfg</i>	Film-based material that has been transferred to 1920x1080 59.94i HD video. The external sync source is NTSC blackburst.
<i>1280x720@5994p_free.cfg</i> <i>1280x720@6000p_free.cfg</i> <i>1920x1035@5994i_free.cfg</i> <i>1920x1035@6000i_free.cfg</i> <i>1920x1080@23976psf_free.cfg</i> <i>1920x1080@24000psf_free.cfg</i> <i>1920x1080@25000psf_free.cfg</i> <i>1920x1080@50i_free.cfg</i> <i>1920x1080@5994i_free.cfg</i> <i>1920x1080@6000i_free.cfg</i>	The corresponding HDTV standard on systems to which no external sync source is connected. These templates are primarily used for 24 fps HD formats. 30 and 25 fps HD formats can be synced to NTSC/PAL blackburst using templates from the two following sections.
<i>1280x720@5994p.cfg</i> <i>1920x1035@5994i.cfg</i> <i>1920x1080@5994i.cfg</i>	The corresponding HDTV standard on systems for which the external sync source is NTSC blackburst.
<i>1920x1080@50i.cfg</i>	The corresponding HDTV standard on systems for which the external sync source is PAL blackburst.
<i>film.cfg</i>	Film-based material.
<i>film_ntsc.cfg</i>	Film-based material that has been transferred to NTSC video. The external sync source is NTSC blackburst.

For information on the HD formats that correspond to these project templates, see [“Supported HD Formats in Project Templates”](#) on page 49.

Onyx 350 with the DG5-2-GVO Board (Graphics Monitor on Channel 1)

On an Onyx 350 with the DG5-2-GVO board and graphics monitor on channel 1, the following devices are connected to the DG5-2-GVO board:

- A graphics monitor is connected to RGB analog channel 1.
- An HD analog RGB broadcast monitor is connected to RGB analog channel 0.

- A 601 SDI broadcast monitor is connected to the 601A link.

You must use an NTSC/PAL sync source to genlock NTSC projects at 59.94 Hz or 50 Hz. For HD projects, you must use the corresponding HD Tri-level sync source. This configuration, as opposed to the previous one, generates a more stable RGB analog HD signal over channel 0.

Use the template from the following table that corresponds to the resolution and sync configuration that you want to work with.

The configuration file template:	Sets parameters for projects to work with:
<i>ntsc.cfg</i>	NTSC 601 material. The external sync source is NTSC blackburst.
<i>pal.cfg</i>	PAL 601 material. The external sync source is PAL blackburst.
<i>24p_ntsc.cfg</i>	Film-based material that has been transferred to NTSC 601 video. The external sync source is NTSC blackburst.
<i>24p_1920x1080@5994i.cfg</i>	Film-based material that has been transferred to 1920x1080 59.94i HD video. The external sync source is NTSC blackburst.
<i>1280x720@5994p.cfg</i> <i>1280x720@6000p.cfg</i> <i>1920x1035@5994i.cfg</i> <i>1920x1035@6000i.cfg</i> <i>1920x1080@23976psf.cfg</i> <i>1920x1080@24000psf.cfg</i> <i>1920x1080@25000psf.cfg</i> <i>1920x1080@50i.cfg</i> <i>1920x1080@5994i.cfg</i> <i>1920x1080@6000i.cfg</i>	The corresponding HDTV standard on systems configured with the DG5-2-GVO board and an HD tri-level sync source. With this configuration, the graphics monitor is connected to channel 1, and the broadcast monitor is connected to channel 0.
<i>film.cfg</i>	Film-based material.
<i>film_ntsc.cfg</i>	Film-based material that has been transferred to NTSC video. The external sync source is NTSC blackburst.

For information on the HD formats that correspond to these project templates, see [“Supported HD Formats in Project Templates”](#) on page 49.

Onyx 350 with the DG5-2-TVO Board (HDGVO)

On an Onyx 350 with the DG5-2-GVO board and HDGVO, the following devices are connected to the DG5-2-TVO board and the video breakout box (VBOB):

- A graphics monitor is connected to RGB analog channel 0 on the DG5-2-TVO board.
- An HD SDI broadcast monitor is connected to the HD Output on the VBOB.
- A 601 SDI broadcast monitor is connected to the SD Output on the VBOB.

There are no sync restrictions. You can use NTSC/PAL or HD Tri-level sync source to genlock SD and HD projects at 59.94 Hz or 50 Hz. As opposed to the previous configuration, which sends an analog RGB signal to the broadcast monitor, the HDGVO option sends HD-SDI output (a 4:2:2 Y'CrCb signal), which is precisely the signal sent to the VTR on output. Alternatively (with the appropriate interface card on your HD broadcast monitor), you can monitor a dual-link HD-SDI 4:4:4 RGB output.

The video breakout box accepts NTSC and PAL sync, which can be used to sync SD-compatible HD formats. For non-SD compatible HD formats, the corresponding HD-Tri-level sync must be fed to the VBOB.

The VBOB then sends sync to the display generator board providing a thoroughly stable SD and HD working environment. The graphics monitor does not have to be connected to channel 1, regardless of the format.

This configuration provides the most stable working environment for HD projects, and is strongly recommended for systems on which a great deal of HD work is done.

Use the template from the following table that corresponds to the resolution and sync configuration that you want to work with.

The configuration file template:	Sets parameters for projects to work with:
<i>ntsc.cfg</i>	NTSC 601 material. The external sync source is NTSC blackburst.
<i>pal.cfg</i>	PAL 601 material. The external sync source is PAL blackburst.
<i>24p_ntsc.cfg</i>	Film-based material that has been transferred to NTSC 601 video. The external sync source is NTSC blackburst.
<i>24p_1920x1080@5994i.cfg</i>	Film-based material that has been transferred to 1920x1080 59.94i HD video. The external sync source is NTSC blackburst.
<i>1280x720@5994p.cfg</i> <i>1280x720@6000p.cfg</i> <i>1920x1080@23976p.cfg</i> <i>1920x1080@23976psf.cfg</i> <i>1920x1080@24000p.cfg</i> <i>1920x1080@24000psf.cfg</i> <i>1920x1080@25000psf.cfg</i> <i>1920x1080@50i.cfg</i> <i>1920x1080@5994i.cfg</i> <i>1920x1080@6000i.cfg</i>	The corresponding standard for systems configured with the DG5-2-TVO board. The external sync source corresponds with the HD format.
<i>film.cfg</i>	Film-based material.
<i>film_ntsc.cfg</i>	Film-based material that has been transferred to NTSC video. The external sync source is NTSC blackburst.

Supported HD Formats in Project Templates

The names of project templates for HD formats contain a string that identifies the format's resolution, frame rate, and scan mode.

Project configuration templates with the following string in their name:	Correspond to the following HD format:
<i>1280x720@5994p</i>	1280x720 progressive scan at 59.94 frames per second (720/59.94p, SMPTE-296M)
<i>1280x720@6000p</i>	1280x720 progressive scan at 60 frames per second (720/60p, SMPTE-296M)
<i>1920x1035@5994i</i>	1920x1035 interlaced scan at 59.94 fields per second (1080/59.94i, SMPTE-240M)
<i>1920x1035@6000i</i>	1920x1035 interlaced scan at 60 fields per second (1080/60i, SMPTE-240M)
<i>1920x1080@23976p</i>	1920x1080 progressive scan at 23.976 frames per second (1080/23.976p, SMPTE-274M, RP-211)
<i>1920x1080@23976psf</i>	1920x1080 progressive segmented frame at 23.976 frames per second (1080/23976psf, SMPTE-274M, RP-211)
<i>1920x1080@24000p</i>	1920x1080 progressive scan at 24 frames per second (1080/24p, SMPTE-274M, RP-211)
<i>1920x1080@24000psf</i>	1920x1080 progressive segmented frames at 24 frames per second (1080/24psf, SMPTE-274M, RP-211)
<i>1920x1080@25000psf</i>	1920x1080 progressive segmented frames at 25 frames per second (1080/25psf, SMPTE-274M, RP-211)
<i>1920x1080@50i</i>	1920x1080 interlaced scan at 50 fields per second (1080/50i, SMPTE-274M)
<i>1920x1080@5994i</i>	1920x1080 interlaced scan at 59.94 fields per second (1080/59.94i, SMPTE-274M)
<i>1920x1080@6000i</i>	1920x1080 interlaced scan at 60 fields per second (1080/60i, SMPTE-274M)

NOTE: If your Tezro workstation is equipped with a DM5 video board, the field dominance at 1080/50i or 1080/24psf resolutions may occasionally be inverted during playback. Only playback is affected—field dominance is not inverted when you output the clip to the VTR. If you experience this problem, contact Customer Support.

Viewing Project Configuration Files

Using a text editor, you can open a project configuration file for viewing. For example, you may want to read the keywords in different project configuration templates to see the differences. This is a good way to learn more about configuration keywords.

To open a project configuration file for viewing:

1. Log in to the application account.
2. In a UNIX shell, type:


```
cd /usr/discreet/project/<project name>/cfg
```
3. Open the project configuration file by typing:


```
nedit <project name>.cfg
```

Note the following as you view the file:

- A pound sign (#) at the beginning of a line indicates a comment. These lines are not parsed during software or project initialisation.
- When you scroll through a file, underlined headings indicate distinct configuration keywords.
- Keyword headings begin with a brief description of the corresponding keyword.
- Keyword headings also include syntax rules for the corresponding keyword followed by examples.
- In some cases, keyword examples are not commented out. In such cases, the keyword examples are default settings. They are parsed during project initialisation and the corresponding parameter is set.

Modifying Project Configuration Files

Project configuration files have open read/write permissions. This means anyone can modify them. In some cases, you may want to change the permissions associated with the files. For example, if you create a custom project configuration file for your project, you may want to restrict access to it using the *chmod* command to prevent others from modifying it. For information on using the *chmod* command, type **man chmod** in a UNIX shell.

Here are some basic guidelines for modifying project configuration files:

- Do not modify the template project configuration files. These are used by the software to generate new project configuration files and should not be altered.
- Create a backup of the project configuration file before making any changes to it.
- Modified project configuration files must end with *_custom.cfg*.

- Although you can modify project configuration files, the Project Management menu provides editing tools to modify projects from session to session as needed. In general, you should not have to modify project configuration files manually.
- You cannot use multiple project configuration files with a single project. A project configuration file must have the same name as its corresponding project and it must be located in the project's `~/cfg` directory. For information on project configuration file keywords, see [“Project Configuration File Keywords”](#) on page 53.

To create a custom project configuration file:

1. Log in as root.
2. In a UNIX shell, go to the directory that contains the desired project `.cfg` file. Type:


```
cd /usr/discreet/project/<project name>/cfg
```
3. Create a copy of the desired project `.cfg` file. Type:


```
cp <template>.cfg <custom_name>_custom.cfg
```

 where `<template>` is the name of the desired project template file and `<custom_name>` is the name of the new, custom project configuration file that you will modify.
4. To open the `<custom_name>_custom.cfg` file in a text editor, type:


```
edit <custom_name>_custom.cfg
```

 A `edit` window appears in which you can view and edit the configuration file.
5. Edit the configuration file as needed. For example, if your location has multiple framestore volumes, you can set a different default volume. To do this, scroll to the “FRAMESTORE” keyword heading.
6. Scroll to the “Example for a Discreet Storage ...” line under the “FRAMESTORE” keyword heading.

The default setting is:

```
FrameStore stonefs
```

In this example, you want to set `stonefs1` as the default volume. The next commented line is:

```
#FrameStore stonefs1
```
7. Uncomment this keyword. To do this, delete the pound sign (`#`) at the beginning of the line.
8. Comment the original setting. To comment a keyword, insert a pound sign (`#`) at the beginning of the line.

The framestore is now set to use `stonefs1` to store clips as the default.

9. Press **CTRL+S** to save the `<custom_name>_custom.cfg` file.

10. Close the *nedit* window.

For more information on project configuration keywords, see [“Project Configuration File Keywords”](#) on page 53.

5

Project Configuration File Keywords

Summary

Introduction	53
Framerate	53
Timecode	54
ColourSpace	54
Hires	55
VideoPreviewWindow	56
FrameStore	56
ArchiveLibrary	57
Environment Directory	57

Introduction

This chapter provides a list of the keywords that appear in a project configuration file. Project configuration files are created using templates that correspond to options you select using the Project Management menu—they rarely require manual modification. In most cases, if you want to edit a project, you can do so using the Edit Project option in the Project Management menu. See the chapter “Managing Projects and Users” in the application User’s Guide.

Framerate

The Framerate keyword sets the default frame rate for the project work session.

The Framerate keyword uses the following syntax:

```
Framerate <frame_rate>
```

where <frame_rate> is 23.976, 24, 25, 29.97, 30, 50, 59.94 or 60 frames per second.

Example of use:

```
Framerate 29.97
```

Timecode

The Timecode keyword sets the default timecode format used by the project.

The Timecode keyword uses the following syntax:

```
Timecode <format>[, DF]
```

Where:	Is:
<format>	One of the following timecode formats: 23.976, 24, 25, 29.97, 30, 50, 59.94, or 60.
DF	An optional parameter indicating drop-frame timecode for 29.97 or 59.94 fps timecode formats.

Examples of use:

Timecode Format	Keyword Example
30 fps	Timecode 30
29.97 fps (drop frame)	Timecode 29.97, DF

ColourSpace

The ColourSpace keyword defines how RGB/YUV conversions are handled. By default, for NTSC or PAL projects, the ITU-R BT.601-5 colour space is used. For DTV projects, the ITU-R BT.709-3 colour space is used.

The ColourSpace keyword uses the following syntax:

```
ColourSpace <space_type>
```

where <space_type> is the colour space type used by the project.

Examples of use:

Project	Keyword Example
NTSC and PAL standard projects (ITU-R BT.601-5)	ColourSpace 601
DTV standard projects (ITU-R BT.709-3)	ColourSpace 709
1920x1035 projects (SMPTE-240M)	ColourSpace 240

Hires

The Hires keyword sets the refresh rate of the graphics monitor. Note the following when setting the refresh rate:

- The rate is measured in Hertz (Hz).
- When using the video board in conjunction with an external sync source, the refresh rate should match the frame rate.
- If you are using a non-integer frame rate (for example, 29.97), set the refresh rate to the nearest integer (in the previous example, 30).
- In some cases, you may want to set the refresh rate using an external utility such as *setmon* or *xsetmon*. You can set the Hires keyword to Custom to override the refresh rate of the application.

The Hires keyword uses the following syntax:

```
Hires [Custom, ]<refresh_rate>[ ,Channel<graphics_display_channel>]
```

Where:	Is:
Custom	An optional parameter that overrides the application's refresh rate. For example, you may choose to override the 60-Hz refresh rate for an NTSC project by using the <code>Custom</code> option with a value of <code>30Hz</code> .
<refresh_rate>	The refresh rate of the graphics monitor. Append <code>Hz</code> for free-running configurations, <code>Hzf</code> for configurations that are fixed to an external sync source.
<graphics_display_channel>	An optional value, usually 1. This value should correspond with the channel on the DCD-2 board to which the SGI graphics monitor is connected.

Examples of use:

Project Resolution	Keyword Example
NTSC	<code>Hires 60Hz</code>
PAL	<code>Hires 50Hz</code>
NTSC (external sync source)	<code>Hires 30Hzf</code>
PAL (external sync source)	<code>Hires 25Hzf</code>
Film	<code>Hires 72Hz</code>
HDTV (with SGI graphics monitor connected to channel 1 on the DG5-2-GVO display generator board and fixed to an external sync source)	<code>Hires 30Hzf, Channel1</code>

Project Resolution	Keyword Example
24p	Hires 72Hz
24p Multi-Master Editing™	Hires 60Hz
Custom	Hires Custom, 30Hz

NOTE: If you are working with film material on an LCD monitor, use the project configuration files *film_48.cfg* or *film_ntsc_48.cfg*. These configuration files contain the correct Hires keyword setting for using an LCD monitor with film material.

VideoPreviewWindow

The VideoPreviewWindow keyword locates a preview device that has a matching size and refresh rate in the VideoPreviewDevice list specified in the *init.cfg* configuration file. When you start the application or switch to another project from the application, the graphics-to-video display is mapped to the appropriate device according to the width, height, and refresh rate specified with this keyword.

The VideoPreviewWindow keyword uses the following syntax:

```
VideoPreviewWindow <xsize>, <ysize>, <refresh_rate>
```

Where:	Is:
<xsize>, <ysize>	The horizontal and vertical size of the area of the image window output to the graphics-to-video display.
<refresh_rate>	The refresh rate of the output.

Example of use:

```
VideoPreviewWindow 720, 486, 30
```

FrameStore

The FrameStore keyword defines the framestore volume to be used for storing clips.

NOTE: Only Autodesk storage is supported.

By default, a project configuration template contains the placeholder <FRAMESTORE> in the place of the framestore definition for the project. When you use the Project Management menu to create a project, the <FRAMESTORE> placeholder is automatically replaced with the framestore defined by the FrameStore keyword in the software initialisation configuration file (*init.cfg*).

The FrameStore keyword uses the following syntax:

```
FrameStore <device>
```

where <device> is the name of the Framestore storing the clips.

Example of use:

```
FrameStore stonefs
```

ArchiveLibrary

Online HTML and ASCII tables of contents will be saved to the filesystem. You can specify the destination directory by setting this keyword. Its default value is */usr/discreet/archive*. You can make the target directory relative to the project directory by prefixing the path with a tilde (e.g. *~/archive*).

NOTE: This keyword, if set, will take precedence over the ArchiveLibrary keyword in the *init.cfg* file.

The ArchiveLibrary keyword uses the following syntax:

```
ArchiveLibrary <directory_path>
```

where <directory_path> is the path to which online HTML and ASCII tables of contents are saved.

Example of use:

```
ArchiveLibrary /usr/discreet/archive
```

Environment Directory

Environment directory keywords specify the paths to directories containing resources for all projects. For example, the LUTs that Autodesk provides are located in the directory */usr/discreet/<product_name>/lut*. You can place resource files in these directories that you want to be able to share between projects more easily.

Not all directories are relevant to the application. However, the inclusion of directories pertaining to other Effects and Editing products allows for cross-product project compatibility.

The ~ in the following table stands for */usr/discreet/<product_name>*.

Resource	Path	Resource File Extension
Action	<i>~/action</i>	<i>.action</i>
Audio	<i>~/audio</i>	<i>.audio</i>
AudioDesk	<i>~/audio</i>	<i>.audiodesk</i>

Resource	Path	Resource File Extension
Audio ImpExp	<i>~/audio</i>	.aiff
AutoMatte	<i>~/automatte</i>	.automatte
AutoPaint	<i>~/paint/autopaint</i>	.auto
Average	<i>~/average</i>	.average
Axis	<i>~/axis</i>	.dve
Batch	<i>~/batch</i>	.batch
BatchClip	<i>~/batchclip</i>	.clip
Blur	<i>~/blur</i>	.blur
Burnin	<i>~/burnin</i>	.burnin
ColourCurves	<i>~/ccurves</i>	.ccurves
ColourFrame	<i>~/colourframe</i>	.col
Colour Warper	<i>~/colourwarper</i>	.cw
Composite	<i>~/composite</i>	.composite
Compound	<i>~/compound</i>	.compound
Correct	<i>~/correct</i>	.correct
Cutout	<i>~/paint/cutout</i>	.sgi
Deal	<i>~/deal</i>	.deal
DeGrain	<i>~/degrain</i>	.degrain
DeInterlace	<i>~/deinterlace</i>	.deinterlace
Desktop	<i>~/desktop</i>	.desk
Difference	<i>~/difference</i>	.diff
Dissolve	<i>~/dissolve</i>	.dissolve
Distort	<i>~/distort</i>	.distort
Documentation	<i>~/documentation</i>	.pdf
DVE	<i>~/dve</i>	.dve
Edge	<i>~/edge</i>	.edge
EDL	<i>~/edl</i>	.edl
Export	<i>~/export</i>	.export
ExprUserFun	<i>~/expressions/userfun</i>	.expressions
FieldMerge	<i>~/fieldmerge</i>	.fieldmerge
Filter	<i>~/filter</i>	.filter
FilmCompress	<i>~/filmcompress</i>	.compress
FilmExpand	<i>~/filmexpand</i>	.expand
Flip	<i>~/flip</i>	.flip
Geometry (Paint)	<i>~/paint/geometry</i>	.geom

Resource	Path	Resource File Extension
GMask	<i>~/gmask</i>	.GMask
Guides	<i>~/guides</i>	.guide
Histo2d	<i>~/histo2d</i>	.histo2d
HotKey	<i>~/hotkey</i>	.hotkey
Image	<i>~/images</i>	See " Image File Extensions " on page 37
Import	<i>~/import</i>	.import
Interlace	<i>~/interlace</i>	.interlace
Key	<i>~/key</i>	.key
Keyer3d	<i>~/keyer3d</i>	.key3d
LensDistort	<i>~/lensDistort</i>	.lensDistort
Letterbox	<i>~/letterbox</i>	.letterbox
LogicOp	<i>~/logicop</i>	.logicop
Lumkey	<i>~/lumkey</i>	.lumkey
Lut	<i>~/lut</i>	.lut
Mask (Paint)	<i>~/paint/mask</i>	.sgi
MasterKeyer	<i>~/masterkeyer</i>	.mkeyer
Mix	<i>~/mix</i>	.mix
Moncal	<i>~/monitor</i>	.monitor
ModularKeyer	<i>~/modularKeyer</i>	.modularKeyer
Mono	<i>~/mono</i>	.mono
Morf	<i>~/morf</i>	.morf
MotionAnalyse	<i>~/motionAnalyse</i>	.motionAnalyse
MotionBlur	<i>~/motionBlur</i>	.motionBlur
NormalMap	<i>~/normalmap</i>	.normalmap
Note	<i>~/note</i>	.note
Omf®	<i>~/images</i>	.omf
Optics	<i>~/optics</i>	.optics
Output	<i>~/output</i>	.output
Paint	<i>~/paint</i>	.paintnode.xml
Picture	<i>~/paint/picture</i>	.sgi
Play	<i>~/play</i>	N/A
Posterize	<i>~/posterize</i>	.posterize
QComp	<i>~/quickcomp</i>	.qcomp
ReGrain	<i>~/regrain</i>	.regrain
Repeat	<i>~/repeat</i>	.repeat

Resource	Path	Resource File Extension
Resize	<i>~/resize</i>	<i>.resize</i>
Restore	<i>~/filmrestore</i>	<i>.restore</i>
Spark	<i>~/sparks</i>	<i>.spark</i>
Stabilizer	<i>~/stabilizer</i>	<i>.stabilizer</i>
Text	<i>~/text</i>	<i>.ttg</i>
Timewarp	<i>~/timewarp</i>	<i>.timewarp</i>
VectorViewer	<i>~/vectorViewer</i>	<i>.vectorViewer</i>
Wipe	<i>~/wipe</i>	<i>.wipe</i>

index

Index

Numerics

3DLUT keyword 27

A

ArchiveLibrary keyword 32, 57

audio file extensions 38

Audiodevice keyword 26

B

BackburnerManagerGroup keyword 40

BackburnerManagerHostname keyword 39

BackburnerManagerPort keyword 39

BackburnerManagerPriority keyword 40

C

CleanerDestinationPath keyword 41

ClipManagementDevice keyword 28

ClipMgtDevice DST keyword 31

ClipMgtDevice File keyword 30

ClipMgtDevice HDCAM keyword 29

ClipMgtDevice Tape keyword 30

ClipMgtDevice VTR keyword 29

ColourSpace keyword 54

conventions, in user's guide 4

D

DefaultPDFViewer keyword 38

DefaultWebBrowser keyword 38

Discreet Native Audio

Audiodevice keyword 26

DiskHealthNotify keyword 15

dlcfg command 8

E

Emulator keyword 25

environment directory keywords 57

F

FontDPSAFM keyword 35

FontDPSBase keyword 34

FontProxyHighString keyword 36

FontProxyLowString keyword 35

Framerate keyword 53

FrameStore keyword 14, 56

G

Gamma keyword 26

H

Hires keyword 55

HiresChannel keyword 20

HtmlLog keyword 34

HtmlTemplates keyword 33

I

image file extensions 37

init.cfg. See software initialisation configuration file

installation

updating configuration file with xdiff 11

L

LogDailyReports keyword 41

M

MaxLibrarySize keyword 31
MemoryApplication keyword 15
Menu keyword 33
MidiDevice keyword 21
Model keyword 33
MoncalDevice keyword 28

N

NetworkPanelDisplay keyword 42

P

project configuration files
 keywords 53
 modifying 50
 template files 43
 viewing 50

S

serial ports
 VTR connections 24
SetupArchiveTape keyword 32
software initialisation configuration file
 cfg directory 7
 keywords 14
 modifying 9
 samples 8
 software initialisation 7
 starting with custom 11
 updating with *xdiff* 11
 viewing 8
 See also configuration files
starting the software
 with a custom configuration file 11

T

TabletDriver keyword 20
TextDefaultFont keyword 34
TextFileEncoding keyword 36
Timecode keyword 54

U

user's guide
 conventions 4

V

Video keyword 17
VideoPreviewDevice keyword 18
VideoPreviewWindow keyword 56
VTR keyword 22