

AUTODESK®  
EFFECTS AND EDITING  
2008

# Configuration File Reference Guide for Linux® 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](http://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](mailto: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 Linux Workstations

Document Version: 1

Date: August 24, 2007

# contents

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
	Summary .....	1
	About Configuration Files.....	1
	About This Guide .....	1
	Related Documentation.....	2
	Notation Conventions .....	4
	Contacting Customer Support .....	4
<b>2</b>	<b>Software Initialisation Configuration File</b>	<b>7</b>
	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
<b>3</b>	<b>Software Initialisation File Keywords</b>	<b>13</b>
	Summary .....	13
	Introduction .....	14
	FrameStore.....	14
	DiskHealthNotify .....	15
	MemoryApplication.....	15

Video.....	17
DviRamp .....	18
VideoPreviewDevice .....	18
TabletDriver.....	20
MidiDevice.....	20
Vtr .....	21
Emulator.....	24
Audiodevice.....	25
Gamma .....	25
3DLUT .....	26
ClipMgtDevice .....	26
MaxLibrarySize .....	29
ArchiveLibrary .....	29
SetupArchiveTape .....	30
Menu.....	30
Model .....	30
HtmlTemplates .....	31
HtmlLog .....	31
TextDefaultFont .....	31
FontDPSBase.....	32
FontDPSAFM .....	32
FontProxyLowString .....	33
FontProxyHighString.....	33
TextFileEncoding .....	34
Image File Extensions.....	34
Audio File Extensions .....	35
DefaultWebBrowser.....	35
DefaultPdfViewer .....	35
BackburnerManagerHostname .....	36
BackburnerManagerPort.....	36
BackburnerManagerPriority.....	37
BackburnerManagerGroup.....	37
CleanerDestinationPath.....	38
LogDailyReports .....	38
NetworkPanelDisplay .....	39

## 4 Project Configuration Files 41

Summary .....	41
About Project Configuration Files .....	41
Project Configuration File Templates .....	41
Viewing Project Configuration Files .....	44

Modifying Project Configuration Files .....	44
<b>5 Project Configuration File Keywords</b>	<b>47</b>
Summary .....	47
Introduction .....	47
Framerate .....	47
Timecode .....	48
ColourSpace.....	48
Hires .....	49
VideoPreviewWindow.....	50
FrameStore.....	50
ArchiveLibrary .....	51
Environment Directory .....	51
<b>Index</b>	<b>55</b>



## Summary

<a href="#">About Configuration Files</a> .....	1
<a href="#">About This Guide</a> .....	1
<a href="#">Related Documentation</a> .....	2
<a href="#">Notation Conventions</a> .....	4
<a href="#">Contacting Customer Support</a> .....	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 kedit.

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 Linux® workstations:

- Autodesk Inferno® 2008, Autodesk Flame® 2008, and Autodesk Flint® 2008
- Autodesk Smoke® 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.

<b>User and Reference Guides</b>	<b>Description</b>
<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

<b>User and Reference Guides</b>	<b>Description</b>
<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
<b>Installation and Configuration Guides</b>	<b>Description</b>
<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
<b>Other Guides</b>	<b>Description</b>
<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](http://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.	<b>rpm -qa</b>
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](http://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](http://www.autodesk.com/resellers).



# Software Initialisation Configuration File



## Summary

<a href="#">About the Software Initialisation Configuration File</a> .....	7
<a href="#">Default Software Initialisation Configuration File</a> .....	7
<a href="#">Sample Software Initialisation Configuration Files</a> .....	8
<a href="#">Viewing a Software Initialisation Configuration File</a> .....	8
<a href="#">Modifying a Software Initialisation Configuration File</a> .....	9
<a href="#">Starting with a Custom Software Initialisation Configuration File</a> .....	11
<a href="#">Updating the Software Initialisation Configuration File</a> .....	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 kedit.

## 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 Linux terminal, 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 Linux terminal.

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 Linux terminal, 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:  
**kedit <custom\_name>.cfg**  
A *kedit* 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 *kedit* 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 Linux terminal, 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 *xxdiff* 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 *xxdiff* 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 *xxdiff* 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 Linux terminal, start *xxdiff* and load the two files for comparison by typing the following:

```
xxdiff <old_file> <new_file>
```

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

```
xxdiff /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 *xxdiff* 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 *xxdiff*.

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 *xxdiff*.

# 3

## Software Initialisation File Keywords

### Summary

<a href="#"><u>Introduction</u></a> .....	14
<a href="#"><u>FrameStore</u></a> .....	14
<a href="#"><u>DiskHealthNotify</u></a> .....	15
<a href="#"><u>MemoryApplication</u></a> .....	15
<a href="#"><u>Video</u></a> .....	17
<a href="#"><u>DviRamp</u></a> .....	18
<a href="#"><u>VideoPreviewDevice</u></a> .....	18
<a href="#"><u>TabletDriver</u></a> .....	20
<a href="#"><u>MidiDevice</u></a> .....	20
<a href="#"><u>Vtr</u></a> .....	21
<a href="#"><u>Emulator</u></a> .....	24
<a href="#"><u>Audiodevice</u></a> .....	25
<a href="#"><u>Gamma</u></a> .....	25
<a href="#"><u>3DLUT</u></a> .....	26
<a href="#"><u>ClipMgtDevice</u></a> .....	26
<a href="#"><u>MaxLibrarySize</u></a> .....	29
<a href="#"><u>ArchiveLibrary</u></a> .....	29
<a href="#"><u>SetupArchiveTape</u></a> .....	30
<a href="#"><u>Menu</u></a> .....	30
<a href="#"><u>Model</u></a> .....	30
<a href="#"><u>HtmlTemplates</u></a> .....	31
<a href="#"><u>HtmlLog</u></a> .....	31
<a href="#"><u>TextDefaultFont</u></a> .....	31
<a href="#"><u>FontDPSBase</u></a> .....	32
<a href="#"><u>FontDPSAFM</u></a> .....	32
<a href="#"><u>FontProxyLowString</u></a> .....	33
<a href="#"><u>FontProxyHighString</u></a> .....	33

<a href="#"><u>TextFileEncoding</u></a> .....	34
<a href="#"><u>Image File Extensions</u></a> .....	34
<a href="#"><u>Audio File Extensions</u></a> .....	35
<a href="#"><u>DefaultWebBrowser</u></a> .....	35
<a href="#"><u>DefaultPdfViewer</u></a> .....	35
<a href="#"><u>BackburnerManagerHostname</u></a> .....	36
<a href="#"><u>BackburnerManagerPort</u></a> .....	36
<a href="#"><u>BackburnerManagerPriority</u></a> .....	37
<a href="#"><u>BackburnerManagerGroup</u></a> .....	37
<a href="#"><u>CleanerDestinationPath</u></a> .....	38
<a href="#"><u>LogDailyReports</u></a> .....	38
<a href="#"><u>NetworkPanelDisplay</u></a> .....	39

## 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 2000 and 4000.

**NOTE:** These RAM amounts apply only to 64-bit versions of the application.

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.

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. The video input devices supported for Linux workstations are AJA\_OEMLH, AJA\_OEM2K, DVS, and DVS\_CENTAURUS.

**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 Video keyword uses the following syntax:

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

Where:	Is:
<video_device>	The video board used by your Linux workstation. It can take the following values: AJA_OEMLH for the AJA OEMLH board used in SD configurations on HP platforms, AJA_OEM2K for the AJA OEM2K board used in HD/2K configurations on HP platforms, DVS for the DVS SDStation board used in SD configurations on IBM® platforms, and DVS_CENTAURUS for the DVS Centaurus board used in HD/2K configurations on IBM platforms.
<default_video_format>	The video input type. For Centaurus boards on IBM 6217 and all AJA boards, the supported video input types are Serial1 and SerialDual. For DVS boards other than the Centaurus board on IBM 6217, the only supported video input type is Serial1.

**Examples of use:**

```
Video AJA_OEMLH, Serial1
```

```
Video AJA_OEM2K, Serial1
```

```
Video DVS, Serial1
```

```
Video DVS_CENTAURUS, Serial1
```

## DviRamp

The DviRamp keyword enables control of the Miranda DVI-Ramp external device. Only available to Linux workstations, the DVI-Ramp outputs a standard definition serial digital video signal (SMPTE-259M-C) or a high definition serial digital video signal (SMPTE-292M).

**NOTE:** Enable this keyword only if you are connecting the Miranda DVI-Ramp external device.

**The DviRamp keyword uses the following syntax:**

```
DviRamp <serial port>, <baud rate>, <parity>, <size>, <stopBits>
```

Where:	Is:
<serial_port>	The type of connection to the DVI-Ramp external device. The values are: ttyS0 for a DVI-Ramp1 connected on a serial port. usb/dviramp for a DVI-Ramp1 connected on a USB port using a USB-Serial adaptor. usb/dviramp2 for DVI-Ramp2 connected on a USB port.
<baud rate>	The baud rate used to communicate with the DVI-Ramp.
<parity>	The parity setting to communicate with the DVI-Ramp. It can be set to EVEN, ODD, or NOPARITY.
<size>	The number of transmitted bits per packet.
<stopBits>	The size of the stop bit.

**Example of use:**

```
DviRamp ttyS0, 38400, none, 8, 1
```

## 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 50.

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 Linux workstations configured with either the Miranda DVI-Ramp external device, or graphics readback (SD timings only) if the hardware is not available.

**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: <code>ntsc</code> , <code>pal</code> , or an HD resolution and timing.
<device>	The graphics-to-video configuration for your system. Set to <code>dviramp</code> or <code>readback</code> .
<channel_number>	The channel on which the signal is output.
<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 corresponding resolution.
<syncsource>	The reference sync. Set to <code>601sync</code> when you are using NTSC or PAL timing and you are connected to an external sync generator. Set to <code>DTVsync</code> when you are using HD timing and you are connected to an external sync generator. Set to <code>reesync</code> 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. Use these optional parameters with NTSC.

**Examples of use:**

```
VideoPreviewDevice ntsc, dviramp, 1, 720, 486, 30, 601sync, 0, 1
```

```
VideoPreviewDevice pal, dviramp, 1, 720, 576, 25, 601sync
```

```
VideoPreviewDevice 1920x1080@50i, dviramp, 1, 1920, 1080, 50i,
```

DTVsync

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 Linux systems without a DV Iramp, 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
```

## TabletDriver

The TabletDriver keyword identifies the tablet driver. Only the Wacom® Intuos-series USB tablets are currently supported for Linux workstations. Connect the tablet to the USB extender which, itself, is connected to the Linux workstation, before booting the workstation and before installing the application. See “Connecting the Tablet” in the chapter “Connecting Peripherals” of the *Hardware Setup Guide* for your workstation.

**The TabletDriver keyword uses the following syntax:**

```
TabletDriver wacom
```

## MidiDevice

The MidiDevice keyword identifies MIDI devices that are used with Autodesk Effects and Editing systems. Only one device is recognized at a 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.

Where:	Is:
<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).

**Example of use:**

```
MidiDevice myMidiDevice, midi_LUCID_ADA8824_A232, /dev/ttyS1,
direct_RS232, 9600, NOPARITY
```

## 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, 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 <code>Serial1</code> to input using a single-link (4:2:2) connection from a device. Set to <code>SerialDual</code> 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).
<colourspace_mode>	The mode that indicates whether colourspace conversion and/or headroom is required for clip I/O. Colourspace conversion with headroom is only available with the DVS Centaurus board. See " <a href="#">Configuring Colourspace Conversion for Device I/O</a> " on page 23.
<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. If using an AJA card, the possible values are: <code>STANDALONE</code> , <code>HOUSE</code> , or <code>DIGITAL1</code> . For a DVS card, use <code>HOUSE</code> or <code>STANDALONE</code> . For a Centaurus card, use <code>HOUSE</code> , <code>STANDALONE</code> , or <code>TRILEVEL</code> .
<serial_port>	The serial port to which the VTR is connected. On the Linux workstation, only <code>Serial2</code> can be specified. It can take the following values: <code>DVS : 0 : 1</code> for Linux workstations with the DVS Centaurus board. <code>DVS : 0 : 2</code> for workstations with the DVS SDStation board. <code>AJA : 0 : 1</code> for workstations using an AJA OEM-2K or an AJA OEM-LH.
<timecode_type>	The timecode type to be returned by the VTR ( <code>Auto</code> , <code>LTC</code> , or <code>VITC</code> ).
<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).

Where:	Is:
<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. NOTE: The value '99' is a special case to be used strictly with HDCAM-SR/720P and AJA. It is not to be used in other circumstances unless specified by 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 BWV50.
<edit_off_delay>	The delay in frames before the edit sync point to send the OFF command. This parameter is optional and it applies only to BWV50.
<vtr_command_delay>	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.

## 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 and Smoke 2K.

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>,
<colorspace_mode>, <output_format>, <output_sync>, <serial_port>,
<emulator_output_delay>, <emulator_input_delay>,
<audio_input_delay>, <audio_output_delay>, <video_precision>
```

Where:	Is:
<name>	The name for the emulator.
<input_format>	The video input format. For a list of the video formats supported by your workstation, see <a href="#">"Video"</a> on page 17.
<timing>	The video resolution and timing of the VTR to be emulated.
<colorspace_mode>	The colourspace mode for video transfers.
<output_format>	The video output format. For a list of the video formats supported by your workstation, see <a href="#">"Video"</a> 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, HOUSE, DVS:0:1,-5, 1, 0.00, 0.00, 8
```

```
Emulator sony, HSDL 1499 sf, SERIALDUAL, 2048x1556_1499SF,
NO_CONVERSION, SERIALDUAL, STANDALONE, DVS:0:1, -6, 1, 0.00,
0.00, 10
```

## Audiodevice

The Audiodevice keyword initialises the specified audio device.

### The Audiodevice keyword uses the following syntax:

```
AudioDevice <type>
```

where <type> is DVS or AJA.

DVS or AJA specifies that the audio subsystem is part of the DVS or AJA audio/video I/O device, respectively. Note that for the audio to work, the Video keyword must be configured with the corresponding video device.

### Example of use:

```
Audiodevice AJA
```

```
Audiodevice DVS
```

## 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, 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
```

## ClipMgtDevice

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

- VTR
- HDCAM

- Tape (SAIT and DTF2 fibre archiving devices only)
- File

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. On Linux workstations, SCSI tape archiving devices are not supported. Use only fibre channel archiving devices, specifically, SAIT and DTF2 devices.

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
DTF2	ClipMgtDevice Tape, /dev/st0, 65536, DTF2
SAIT	ClipMgtDevice Tape, /dev/st0, 65536, SAIT

**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.

**Example of use:**

```
ClipMgtDevice File, /usr/tmp/archive, 1024
```

## 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 Linux terminal:

```
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.

**Example of use:**

```
SetupArchiveTape /usr/tmp/setups.tar
```

## 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, Flame, and Flint only.

**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 `<font>` 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>code</code>	The Unicode value associated with the character that you want to display.
<code>...</code>	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>code</code>	The Unicode value associated with the character that you want to display.
<code>...</code>	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 Linux terminal.

**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

Image Format	Extension
Photoshop®	psd
OpenEXR	exr
Quicktime®	mov

## 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

## 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 firefox
```

## 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
```

### 3 Software Initialisation File Keywords

# Project Configuration Files



## Summary

<a href="#">About Project Configuration Files</a> .....	41
<a href="#">Project Configuration File Templates</a> .....	41
<a href="#">Viewing Project Configuration Files</a> .....	44
<a href="#">Modifying Project Configuration Files</a> .....	44

## 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 kedit. 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 44
- [“Modifying Project Configuration Files”](#) on page 44

## 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 44.

Select the template from the following table that corresponds to the resolution of your project.

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>film.cfg</code>	Film-based material.
<code>film_48.cfg</code>	Film-based material that is viewed on the Sony™ SDM-P234/B widescreen LCD monitor. If <code>film.cfg</code> is used instead, the monitor will be unable to display the signal.
<code>film_ntsc.cfg</code>	Film-based material that has been transferred to NTSC video. The external sync source is NTSC blackburst.
<code>film_ntsc_48.cfg</code>	Film-based material that has been transferred to NTSC video and is viewed on the Sony SDM-P234/B widescreen LCD monitor. The external sync source is NTSC blackburst. If <code>film_ntsc.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>1280x720@5994p_free.cfg</i> <i>1280x720@6000p_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 HD standard on Linux workstations with the DVS Centaurus board, but to which no external sync source is connected.
<i>1280x720@5994p.cfg</i> <i>1280x720@6000p.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 HD standard on Linux workstations with the DVS Centaurus board. The external sync source corresponds with the HD format.

**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.

## 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>1920x1080@23976psf</i>	1920x1080 progressive segmented frame at 23.976 frames per second (1080/23976psf, 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)

Project configuration templates with the following string in their name:	Correspond to the following HD format:
<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)

## 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 Linux terminal, type:
 

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

```
kedit <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 Linux terminal.

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 47.

#### To create a custom project configuration file:

1. Log in as root.
2. In a Linux terminal, 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:
 

```
kedit <custom_name>_custom.cfg
```

 A `kedit` 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 *keedit* window.

For more information on project configuration keywords, see [“Project Configuration File Keywords”](#) on page 47.

# 5

## Project Configuration File Keywords

### Summary

<a href="#">Introduction</a> .....	47
<a href="#">Framerate</a> .....	47
<a href="#">Timecode</a> .....	48
<a href="#">ColourSpace</a> .....	48
<a href="#">Hires</a> .....	49
<a href="#">VideoPreviewWindow</a> .....	50
<a href="#">FrameStore</a> .....	50
<a href="#">ArchiveLibrary</a> .....	51
<a href="#">Environment Directory</a> .....	51

### 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:

- If you are using DVS, you must set the refresh rate equal to the video rate using Hzf (Hertz framelock).
- 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).

### The Hires keyword uses the following syntax:

Hires [Custom,]<refresh\_rate>

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 Custom option with a value of 30Hz.
<refresh_rate>	The refresh rate of the graphics monitor. Append Hz for free-running configurations, Hzf for configurations that are fixed to an external sync source.

### Examples of use:

Project Resolution	Keyword Example
NTSC	Hires 60Hz
PAL	Hires 50Hz
NTSC (external sync source)	Hires 30Hzf
PAL (external sync source)	Hires 25Hzf
Film	Hires 72Hz
24p	Hires 72Hz
24p Multi-Master Editing™	Hires 60Hz
Custom	Hires Custom, 30Hz

## VideoPreviewWindow

The VideoPreviewWindow keyword is used when using a Miranda device. It 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	<code>~/action</code>	<code>.action</code>
Audio	<code>~/audio</code>	<code>.audio</code>
AudioDesk	<code>~/audio</code>	<code>.audiodesk</code>
Audio ImpExp	<code>~/audio</code>	<code>.aiff</code>
AutoMatte	<code>~/automatte</code>	<code>.automatte</code>
AutoPaint	<code>~/paint/autopaint</code>	<code>.auto</code>
Average	<code>~/average</code>	<code>.average</code>
Axis	<code>~/axis</code>	<code>.dve</code>
Batch	<code>~/batch</code>	<code>.batch</code>
BatchClip	<code>~/batchclip</code>	<code>.clip</code>

Resource	Path	Resource File Extension
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
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 " <a href="#">Image File Extensions</a> " on page 34
Import	<i>~/import</i>	.import
Interlace	<i>~/interlace</i>	.interlace

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

**5** Project Configuration File Keywords

<b>Resource</b>	<b>Path</b>	<b>Resource File Extension</b>
Wipe	<i>~/wipe</i>	<i>.wipe</i>

# index

## Index

### Numerics

3DLUT keyword 26

### A

ArchiveLibrary keyword 29, 51

audio file extensions 35

Audiodevice keyword 25

### B

BackburnerManagerGroup keyword 37

BackburnerManagerHostname keyword 36

BackburnerManagerPort keyword 36

BackburnerManagerPriority keyword 37

### C

CleanerDestinationPath keyword 38

ClipManagementDevice keyword 26

ClipMgtDevice File keyword 28

ClipMgtDevice HDCAM keyword 27

ClipMgtDevice Tape keyword 28

ClipMgtDevice VTR keyword 27

ColourSpace keyword 48

conventions, in user's guide 4

### D

DefaultPDFViewer keyword 35

DefaultWebBrowser keyword 35

DiskHealthNotify keyword 15

dlcfg command 8

DviRamp keyword 18

### E

Emulator keyword 24

environment directory keywords 51

### F

FontDPSAFM keyword 32

FontDPSBase keyword 32

FontProxyHighString keyword 33

FontProxyLowString keyword 33

Framerate keyword 47

FrameStore keyword 14, 50

### G

Gamma keyword 25

### H

Hires keyword 49

HtmlLog keyword 31

HtmlTemplates keyword 31

### I

image file extensions 34

init.cfg. See software initialisation configuration file

installation

    updating configuration file with xdiff 11

### L

LogDailyReports keyword 38

### M

MaxLibrarySize keyword 29

MemoryApplication keyword 15

Menu keyword 30

MidiDevice keyword 20  
Model keyword 30

## N

NetworkPanelDisplay keyword 39

## P

project configuration files  
    keywords 47  
    modifying 44  
    template files 41  
    viewing 44

## S

SetupArchiveTape keyword 30  
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 31  
TextFileEncoding keyword 34  
Timecode keyword 48

## U

user's guide  
    conventions 4

## V

Video keyword 17  
VideoPreviewDevice keyword 18  
VideoPreviewWindow keyword 50  
VTR keyword 21