

Autodesk®

Visual Effects and Finishing
2009 Extension 1 Edition

Configuration File Reference Guide

© 2008 Autodesk Canada Co. All rights reserved. Except as otherwise permitted by Autodesk Canada Co., 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 relating to MXF-SDK were developed by Media, Objects and Gadgets – Soluções de Software e Hardware, S.A. (<http://www.mog-solutions.com>) in co-operation with Institut für Rundfunktechnik GmbH (<http://www.irt.de>).

Portions relating to JPEG © Copyright 1991-1998 Thomas G. Lane. All rights reserved. This software is based in part on the work of the Independent JPEG Group.

Portions relating to JPEG are copyright © 1991-1996, Thomas G. Lane. All Rights Reserved except as specified below. The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

Portions relating to LAPACK v. 3 © 1999. LAPACK Users' Guide Third Edition is the official reference for LAPACK.

Portions relating to BLAS v. 3 © 2005. ACM Transactions on Mathematical Software.

Portions relating to MD5 Copyright © 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved. License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function. License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work. RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind. These notices must be retained in any copies of any part of this documentation and/or software.

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

Portions relating to GLEE v. 5.21 Copyright © 2006 Ben Woodhouse. 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 as the first lines of this file unmodified. 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. THIS SOFTWARE IS PROVIDED BY BEN WOODHOUSE "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 BEN WOODHOUSE 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 powered by Automatic Duck. © 2006 Automatic Duck, Inc. All rights reserved.

Portions relating to Glibc v. 2.3.4 file contains the copying permission notices for various files in the GNU C Library distribution that have copyright owners other than the Free Software Foundation. These notices all require that a copy of the notice be included in the accompanying documentation and be distributed with binary distributions of the code, so be sure to include this file along with any binary distributions derived from the GNU C Library.

Portions relating to Firefox 2 are © 2005-2008. Mozilla. All rights reserved. Distributed under the Mozilla Public License Version 1.1.

Portions relating to GnuPlot v. 4.0 Copyright ©1986 - 1993, 1998, 2004 Thomas Williams, Colin Kelley. 1. distribute the corresponding source modifications from the released version in the form of a patch file along with the binaries, 2. add special version identification to distinguish your version in addition to the base release version number, 3. provide your name and address as the primary contact for the support of your modified version, and 4. retain our contact information in regard to use of the base software.

Portions relating to "dslib" C/C++ Copyright 1988-1989 Eugene Dronek and Rich Morin

Portions relating to TiC++ v. 1.0 Copyright 2006 Ryan Puzstai, Ryan Mulder

Portions relating to NumPy v. 1.0.4 Copyright ©2005, NumPy Developers. 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 the NumPy Developers nor the names of any contributors 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 PostgreSQL v. 7.4.8-1 Copyright ©1996-2003, PostgreSQL Global Development Group.

Portions relating to PostgreSQL v. 7.4.8-1 Copyright © 1994, The Regents of the University of California. Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies. IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Portions relating to PostgreSQL Server v. 7.4.8-1 Copyright ©1996-2003, PostgreSQL Global Development Group.

Portions relating to PostgreSQL Server v. 7.4.8-1 Copyright © 1994, The Regents of the University of California. Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies. IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Portions relating to Mesa v. 6.5.1 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 pySerial Copyright © 2001-2004 Chris Liechti cliechti@gmx.net; All Rights Reserved.

Portions relating to OpenExr v. 1.2.1 Copyright © 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 libtiff are Copyright © 1988-1997 Sam Leffler. Copyright © 1991-1997 Silicon Graphics, Inc. Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics. THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND,

OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Portions relating to MPEG Layer- 3, supply of this product does not convey a license under the relevant intellectual property of Thomson multimedia and/or Fraunhofer Gesellschaft nor imply any right to use this product in any finished end user or ready-to-use final product. An independent license for such use is required. For details, please visit <http://www.mp3licensing.com>.

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, ADI, Alias, Alias (swirl design/logo), AliasStudio, AliasWavefront (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, DesignStudio (design/logo), Design Web Format, DWF, DWG, DWG (logo), DWG Extreme, DWG TrueConvert, DWG TrueView, DXF, Ecotect, Exposure, Extending the Design Team, FBX, Filmbox, FMDesktop, Freewheel, GDX Driver, Gmax, Green Building Studio, Heads-up Design, Heidi, HumanIK, IDEA Server, i-drop, ImageModeler, iMOUT, Incinerator, Inventor, Inventor LT, Kaydara, Kaydara (design/logo), Kynapse, Kynogon, LocationLogic, Lustre, Matchmover, Maya, Mechanical Desktop, MotionBuilder, Movimento, Mudbox, NavisWorks, ObjectARX, ObjectDBX, Open Reality, Opticore, Opticore Opus, PolarSnap, PortfolioWall, Powered with Autodesk Technology, Productstream, ProjectPoint, ProMaterials, RasterDWG, Reactor, RealDWG, Real-time Roto, REALVIZ, Recognize, Render Queue, Retimer, Reveal, Revit, Showcase, ShowMotion, SketchBook, SteeringWheels, Stitcher, StudioTools, Topobase, Toxik, TrustedDWG, ViewCube, Visual, Visual Construction, Visual Drainage, Visual Landscape, Visual Survey, Visual Toolbox, Visual LISP, Voice Reality, Volo, Vtour, Wiretap, and WiretapCentral.

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, and Wire.

The following are registered trademarks or trademarks of Moldflow Corp. in the USA and/or other countries: Moldflow MPA, MPA (design/logo), Moldflow Plastics Advisers, MPI, MPI (design/logo), Moldflow Plastics Insight, MPX, MPX (design/logo), Moldflow Plastics Xpert.

Automatic Duck and the duck logo are trademarks of Automatic Duck, Inc. FFmpeg is a trademark of Fabrice Bellard, originator of the FFmpeg project. 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 Visual Effects and Finishing 2009 Extension 1 Edition Configuration File Reference Guide
Document Version: 1
Date: September 26, 2008

Contents

Chapter 1	Introduction	1
	About Configuration Files	1
	About This Guide	2
	Related Documentation	2
	Notation Conventions	2
	Contacting Customer Support	3
Chapter 2	Software Initialisation Configuration File	5
	About the Software Initialisation Configuration File	5
	Default Software Initialisation Configuration File	6
	Sample Software Initialisation Configuration Files	6
	Viewing a Software Initialisation Configuration File	7
	Modifying a Software Initialisation Configuration File	8
	Starting with a Custom Software Initialisation Configuration File	10
	Updating the Software Initialisation Configuration File	10
Chapter 3	Software Initialisation File Keywords	13
	Introduction	14
	DiskHealthNotify	14
	MemoryApplication	15
	Working with 4K 12-bit Film	15
	Cannot Allocate Memory Buffer Messages	16

Memory Panic	16
Video	16
DviRamp	17
VideoPreviewDevice	18
Configuring Readback Mode	20
TabletDriver	20
MidiDevice	20
Vtr	21
Configuring Colourspace Conversion for Device I/O	24
Emulator	25
Audiodevice	26
ClipMgtDevice	27
ClipMgtDevice VTR	27
ClipMgtDevice HDCAM	27
ClipMgtDevice Tape	28
ClipMgtDevice File	29
MaxLibrarySize	29
ArchiveLibrary	30
SetupArchiveTape	30
Menu	31
Model	31
HtmlTemplates	31
HtmlLog	32
TextDefaultFont	32
FontDPSBase	32
FontDPSAFM	33
FontProxyLowString	33
FontProxyHighString	34
TextFileEncoding	35
Image File Extensions	35
Audio File Extensions	36
DefaultWebBrowser	37
DefaultPdfViewer	37
BackburnerManagerHostname	38
BackburnerManagerPort	38
BackburnerManagerPriority	39
BackburnerManagerGroup	39
BackburnerManagerGroupCapability	40
CleanerDestinationPath	41
LogDailyReports	41
NetworkPanelDisplay	42

Chapter 4 Project Configuration Files 43

About Project Configuration Files	43
Project Configuration File Templates	44
Supported HD Formats in Project Templates	46

	Viewing Project Configuration Files	46
	Modifying Project Configuration Files	47
Chapter 5	Project Configuration File Keywords	49
	Introduction	49
	Framerate	50
	Timecode	50
	ColourSpace	50
	Hires	51
	VideoPreviewWindow	52
	ArchiveLibrary	53
	Environment Directory	53
	Index	59

Introduction



Topics in this chapter:

- [About Configuration Files](#) on page 1
- [About This Guide](#) on page 2
- [Related Documentation](#) on page 2
- [Notation Conventions](#) on page 2
- [Contacting Customer Support](#) on page 3

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® Visual Effects and Finishing application through its software initialisation and project configuration files. Its content pertains to the following Autodesk applications for Linux® workstations:

- Autodesk Inferno® 2009, Autodesk Flame® 2009, and Autodesk Flint® 2009
- Autodesk Smoke® 2009, and Autodesk Backdraft® Conform 2009

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

Related Documentation

Documentation for this release is installed with the product as PDF files and as an HTML help system, and is also available on the Autodesk web site at <http://www.autodesk.com/me-documentation>. From this page you can access the complete documentation library.

You should also refer to the product release notes for all late-breaking release information.

Notation Conventions

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

Convention	Example
Text that you enter in a command line or shell appears in Courier bold. You must press the Enter key after each command.	rpm -qa
Variable names appear in Courier, enclosed in angle brackets. No spaces are allowed in variable names	<variable_name>
Variables that appear enclosed in square brackets are optional.	[<filename>]

Convention	Example
Feedback from the command line or shell appears in Courier.	<code>limit coredump- size</code>
Directory names, filenames, URLs, and command line utilities appear in italics.	<i>/usr/discreet</i>

Contacting Customer Support

For Autodesk Media and Entertainment Customer Support, visit <http://www.autodesk.com/support>.

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

Software Initialisation Configuration File

2

Topics in this chapter:

- [About the Software Initialisation Configuration File](#) on page 5
- [Default Software Initialisation Configuration File](#) on page 6
- [Sample Software Initialisation Configuration Files](#) on page 6
- [Viewing a Software Initialisation Configuration File](#) on page 7
- [Modifying a Software Initialisation Configuration File](#) on page 8
- [Starting with a Custom Software Initialisation Configuration File](#) on page 10
- [Updating the Software Initialisation Configuration File](#) on page 10

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 10.

Sample Software Initialisation Configuration Files

The Visual Effects and Finishing application installation packages come with sample software initialisation configuration files for 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 based on 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 directory `/usr/discreet/<product_home>/cfg/<platform>/sample`. 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:

<code>init_film.cfg</code>	Initialise the software to work with film projects.
<code>init_video.cfg</code>	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 7
- [Modifying a Software Initialisation Configuration File](#) on page 8

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:
`dlcfg`

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 10.

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 Open the *<custom_name>.cfg* file in a text editor.
- 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 “CLIPMGTDVICE” 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 “CLIPMGTDVICE” 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 Delete the pound sign at the beginning of the example line to uncomment it.

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 Save and close the *<custom_name>.cfg* file.

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
```

TIP 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*.

Software Initialisation

File Keywords

3

Topics in this chapter:

- [Introduction](#) on page 14
- [DiskHealthNotify](#) on page 14
- [MemoryApplication](#) on page 15
- [Video](#) on page 16
- [DviRamp](#) on page 17
- [VideoPreviewDevice](#) on page 18
- [TabletDriver](#) on page 20
- [MidiDevice](#) on page 20
- [Vtr](#) on page 21
- [Emulator](#) on page 25
- [Audiodevice](#) on page 26
- [ClipMgtDevice](#) on page 27
- [MaxLibrarySize](#) on page 29
- [ArchiveLibrary](#) on page 30
- [SetupArchiveTape](#) on page 30
- [Menu](#) on page 31
- [Model](#) on page 31
- [HtmlTemplates](#) on page 31
- [HtmlLog](#) on page 32
- [TextDefaultFont](#) on page 32
- [FontDPSBase](#) on page 32

- [FontDPSAFM](#) on page 33
- [FontProxyLowString](#) on page 33
- [FontProxyHighString](#) on page 34
- [TextFileEncoding](#) on page 35
- [Image File Extensions](#) on page 35
- [Audio File Extensions](#) on page 36
- [DefaultWebBrowser](#) on page 37
- [DefaultPdfViewer](#) on page 37
- [BackburnerManagerHostname](#) on page 38
- [BackburnerManagerPort](#) on page 38
- [BackburnerManagerPriority](#) on page 39
- [BackburnerManagerGroup](#) on page 39
- [BackburnerManagerGroupCapability](#) on page 40
- [CleanerDestinationPath](#) on page 41
- [LogDailyReports](#) on page 41
- [NetworkPanelDisplay](#) on page 42

Introduction

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

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

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.

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 Finishing applications or 700 MB for Visual 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 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. 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 using the Miranda DVI-Ramp external device. The DviRamp keyword is automatically commented out if the application installation script detects an NVIDIA® SDI board connected to the system.

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: <code>ttyS0</code> for a DVI-Ramp1 connected on a serial port. <code>usb/dviramp</code> for a DVI-Ramp1 connected on a USB port using a USB-Serial adaptor. <code>usb/dviramp2</code> 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 52.

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 workstations configured with either a Miranda DVI-Ramp external device, a NVIDIA SDI board, 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> (if you are using a DVI ramp), <code>nvidia</code> (if you are using an NVIDIA SDI board), 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, nvidia, 1, 720, 576, 25, 601sync
```

```
VideoPreviewDevice 1920x1080@50i, nvidia, 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, nvidia, 1, 1280,  
720, 5994p, 60lsync
```

Configuring Readback Mode

On systems without a DVIramp or a NVIDIA SDI board, you must enable the following lines to ensure video preview using the readback mode:

```
VideoPreviewDevice pal, readback, 2, 720, 576, 25,  
60lsync
```

```
VideoPreviewDevice ntsc, readback, 2, 720, 486, 30,  
60lsync
```

TabletDriver

The TabletDriver keyword identifies the tablet driver. Only Wacom® Intuos-series USB tablets are currently supported. Connect the tablet to the USB extender which, itself, is connected to the workstation, before booting the workstation and before installing the application. See 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 Visual Effects and Finishing 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.
<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 Serial1 to input using a single-link (4:2:2) connection from a device. Set to SerialDual to input using a dual-link (4:4:4) connection from a device.
<timing>	The I/O timing associated with the video standard of the VTR (NTSC, PAL, or HD, if applicable).
<colorspace_mode>	The mode that indicates whether colourspace conversion and/or headroom is required for clip I/O. Colourspace conversion with headroom is only available with the DVS Centaurus board. See Configuring Colourspace Conversion for Device I/O on page 24.
<output_format>	The video output type. Set to Serial1 to output using a single-link (4:2:2) connection to a device. Set to SerialDual 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: STANDALONE, HOUSE, or DIGITAL1. For a DVS card, use HOUSE or STANDALONE. For a Centaurus card, use HOUSE, STANDALONE, or TRILEVEL.
<serial_port>	The serial port to which the VTR is connected. On the Linux workstation, only Serial2 can be specified. It can take the following values: DVS : 0 : 1 for Linux workstations with the DVS Centaurus board. DVS : 0 : 2

Where:	Is:
	for workstations with the DVS SDStation board. AJA : 0 : 1 for workstations using an AJA OEM-2K or an AJA OEM-LH.
<timecode_type>	The timecode type to be returned by the VTR (Auto, LTC, or VITC).
<video_output_delay>	Video output delay in frames.
<video_input_delay>	Video input delay in frames.
<pre_roll>	Pre-roll in frames or seconds: Use integers to specify pre-roll in frames. Use decimals to specify pre-roll in seconds.
<post_roll>	Post-roll in frames or seconds: Use integers to specify post-roll in frames. Use decimals to specify post-roll in seconds.
<audio_input_delay>	The offset value to have sync audio with video on input. Integer units represent frames; this delay should be 0 by default.
<audio_output_delay>	The offset value to have sync audio with video on output. Integer units represent frames; this delay should be 0 by default.
<video_precision>	The precision of the video interface (8 or 10 bits).
<cueup_mode>	The method by which the VTR is cued. This parameter is optional. Use <code>vtrcueing</code> to make the application use the VTR Cue command directly to cue the VTR. Use <code>vtrfff</code> if using <code>vtrcueing</code> causes the VTR to react slowly when cued, such as with a Betacam SP™.
<TC_transition_delay>	The delay in milliseconds after a vertical sync, before requesting the VTR timecode. This value should only be set or changed with the help of technical support. 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.

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

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

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 Video on page 16.
<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 Video on page 16.
<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.

Where:	Is:
<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 Audio device 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
```

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 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.

The HtmlLog keyword uses the following syntax

```
HtmlLog <directory_path>
```

where <directory_path> is the destination directory.

Example of use

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

TextDefaultFont

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

The TextDefaultFont keyword uses the following syntax

```
TextDefaultFont <font>
```

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

Example of use

```
TextDefaultFont Discreet
```

FontDPSBase

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

The FontDPSBase keyword uses the following syntax

FontDPSBase <directory_path>

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

Example of use

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

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

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

FontDPSAFM

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

The FontDPSAFM keyword uses the following syntax

```
FontDPSAFM <directory_path>
```

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

Example of use

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

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

FontProxyLowString

The FontProxyLowString keyword, along with the FontProxyHighString keyword, specifies which characters to draw in font proxies. By default, the

proxy string is “Aa”. You can override the default string for non-extended character sets by uncommenting and editing the `FontProxyLowString` keyword option.

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

The `FontProxyLowString` keyword uses the following syntax

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

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

Example of use

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

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

FontProxyHighString

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

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

The `FontProxyHighString` keyword uses the following syntax

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

Where: **Is:**

`code` The Unicode value associated with the character that you want to display.

`...` Up to four (for a total of five) more Unicode values for the font proxy string.

Example of use

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

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

TextFileEncoding

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

Image Format	Extension
Digital Picture Exchange	dpx
Jpeg	jpg
Pict	pict
Pixar	picio
Sgi®	sgi
Softimage®	pic
Targa®	tga
Maya®	iff
Tiff	tif
Wavefront®	rla
Photoshop®	psd
OpenEXR	exr
Quicktime®	mov
MXF	mxp

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

Audio Format	Extension
Microsoft®	wav
Berkeley (BSD)	bsf
AVR (Audio Visual Research)	avr
MPEG-1 Layer3	mp3

DefaultWebBrowser

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

The DefaultWebBrowser keyword uses the following syntax

```
DefaultWebBrowser <browser>
```

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

Example of use

```
DefaultWebBrowser 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 or Adobe® Reader™.

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 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.
- 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 sets the port that your application uses to submit jobs directly to a Burn™ render node, bypassing the Backburner Manager.

This keyword is used only in special situations, for example when testing Autodesk Developer Network Sparks® plug-ins. Uncomment this keyword only if you want to submit jobs directly to a Burn render node. See the *Sparks API Reference Guide* and the *Burn Installation and User Guide*.

The BackburnerManagerPort keyword uses the following syntax

```
BackburnerManagerPort <port_number>
```

By default, <port_number> is set to 9000, which is the default port where Burn accepts jobs from workstations on the network. Do not change this value unless you are sure that the Burn render node is using a different port number.

Example of use

BackburnerManagerPort 9000

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
```

BackburnerManagerGroupCapability

The `BackburnerManagerGroupCapability` keyword specifies whether the nodes in your rendering network are equipped with GPU-accelerated graphics boards or not. Based on the value of this keyword, the Visual Effects and Finishing application enables or disables the submission of jobs that require a GPU (such as floating point jobs) to the rendering network.

The `BackburnerManagerGroupCapability` keyword uses the following syntax

```
BackburnerManagerGroupCapability <group_capability>
```

where `<group_capability>` can be `software`, `gpu`, or `hybrid` depending on the hardware of the nodes in the rendering network.

- If none of the nodes in your rendering network are equipped with GPUs, set the value of the keyword to `software`. The application does not send jobs that require a GPU to the rendering network, but only jobs that can be rendered in software mode (using OSMesa) by the render nodes.
- If ALL the nodes in your rendering network are equipped with GPUs, set the value of the keyword to `gpu`. The application sends all jobs to the GPU-equipped nodes, even if some jobs do not specifically require a GPU render node. The GPU-equipped render nodes will render jobs that require a GPU, as well as OSMesa jobs.

NOTE If your rendering network also contains render nodes without a GPU, but the `BackburnerManagerGroupCapability` keyword is incorrectly set to `gpu`, all jobs are sent only to GPU-equipped render nodes, and the nodes without a GPU are never used.

- If your rendering network contains a mix of nodes with GPUs and without GPUs, set the keyword to `hybrid`. The application sends all jobs to the rendering network, and Backburner Manager distributes each job to the appropriate type of render node. Jobs that require a GPU are sent only to GPU-equipped render nodes, while jobs that do not require a GPU are sent to any available render node (GPU or non-GPU) to be rendered in software mode.

NOTE Set the `BackburnerManagerGroupCapability` keyword to `gpu` or `hybrid` only if you are sure that at least one node in your rendering network is equipped with a GPU. Attempting to submit a job that requires a GPU to a rendering network with no GPU-equipped render node results in the job being stuck in the rendering queue indefinitely.

Example of use

```
BackburnerManagerGroupCapability software
```

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

Example of use

```
NetworkPanelDisplay ShowMounted
```

Project Configuration Files

4

Topics in this chapter:

- [About Project Configuration Files](#) on page 43
- [Project Configuration File Templates](#) on page 44
- [Viewing Project Configuration Files](#) on page 46
- [Modifying Project Configuration Files](#) on page 47

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 46

- [Modifying Project Configuration Files](#) on page 47

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.

TIP 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 47.

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.

The configuration file template:	Sets parameters for projects to work with:
<i>24p_1920x1080@5994i.cfg</i>	Film-based material that has been transferred to 1920x1080 59.94i HD video. The external sync source is NTSC blackburst.
<i>film.cfg</i>	Film-based material.
<i>film_48.cfg</i>	Film-based material that is viewed on the Sony™ SDM-P234/B widescreen LCD monitor. If <i>film.cfg</i> is used instead, the monitor will be unable to display the signal.
<i>film_ntsc.cfg</i>	Film-based material that has been transferred to NTSC video. The external sync source is NTSC blackburst.
<i>film_ntsc_48.cfg</i>	Film-based material that has been transferred to NTSC video and is viewed on the Sony SDM-P234/B widescreen LCD monitor. The external sync source is NTSC blackburst. If <i>film_ntsc.cfg</i> is used instead, the monitor will be unable to display the signal.
<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)
<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
```

Open the project configuration file by typing:

```
ked it <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 49.

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 Open the `<custom_name>_custom.cfg` file in a text editor.
- 5 Edit the configuration file as needed, and save it.

For more information on project configuration keywords, see [Project Configuration File Keywords](#) on page 49.

Project Configuration File Keywords

5

Topics in this chapter:

- [Introduction](#) on page 49
- [Framerate](#) on page 50
- [Timecode](#) on page 50
- [ColourSpace](#) on page 50
- [Hires](#) on page 51
- [VideoPreviewWindow](#) on page 52
- [ArchiveLibrary](#) on page 53
- [Environment Directory](#) on page 53

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 your application User 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 48Hz
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
```

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 Visual Effects and Finishing 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>

Resource	Path	Resource File Extension
AutoPaint	<i>~/paint/autopaint</i>	.auto
Average	<i>~/average</i>	.average
Axis	<i>~/axis</i>	.dve
Batch	<i>~/batch</i>	.batch
BatchClip	<i>~/batchclip</i>	.clip
Blur	<i>~/blur</i>	.blur
Burnin	<i>~/burnin</i>	.burnin
ColourCurves	<i>~/ccurves</i>	.ccurves
ColourFrame	<i>~/colourframe</i>	.col
Colour Warper	<i>~/colourwarper</i>	.cw
Composite	<i>~/composite</i>	.composite
Compound	<i>~/compound</i>	.compound
Correct	<i>~/correct</i>	.correct
Cutout	<i>~/paint/cutout</i>	.sgi
Deal	<i>~/deal</i>	.deal
DeGrain	<i>~/degrain</i>	.degrain
DeInterlace	<i>~/deinterlace</i>	.deinterlace
Desktop	<i>~/desktop</i>	.desk
Difference	<i>~/difference</i>	.diff
Dissolve	<i>~/dissolve</i>	.dissolve
Distort	<i>~/distort</i>	.distort
Documentation	<i>~/documentation</i>	.pdf
DVE	<i>~/dve</i>	.dve

Resource	Path	Resource File Extension
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 Image File Extensions on page 35
Import	<i>~/import</i>	.import
Interlace	<i>~/interlace</i>	.interlace
Key	<i>~/key</i>	.key
Keyer3d	<i>~/keyer3d</i>	.key3d
LensDistort	<i>~/lensDistort</i>	.lensDistort
Letterbox	<i>~/letterbox</i>	.letterbox
LogicOp	<i>~/logicop</i>	.logicop

Resource	Path	Resource File Extension
Lumkey	<i>~/lumkey</i>	.lumkey
Lut	<i>~/lut</i>	.lut
Mask (Paint)	<i>~/paint/mask</i>	.sgi
MasterKeyer	<i>~/masterkeyer</i>	.mkeyer
Mix	<i>~/mix</i>	.mix
Moncal	<i>~/monitor</i>	.monitor
ModularKeyer	<i>~/modularKeyer</i>	.modularKeyer
Mono	<i>~/mono</i>	.mono
Morf	<i>~/morf</i>	.morf
MotionAnalyse	<i>~/motionAnalyse</i>	.motionAnalyse
MotionBlur	<i>~/motionBlur</i>	.motionBlur
NormalMap	<i>~/normalmap</i>	.normalmap
Note	<i>~/note</i>	.note
Omf [®]	<i>~/images</i>	.omf
Optics	<i>~/optics</i>	.optics
Output	<i>~/output</i>	.output
Paint	<i>~/paint</i>	.paintnode.xml
Picture	<i>~/paint/picture</i>	.sgi
Play	<i>~/play</i>	N/A
Posterize	<i>~/posterize</i>	.posterize
QComp	<i>~/quickcomp</i>	.qcomp
ReGrain	<i>~/regrain</i>	.regrain
Repeat	<i>~/repeat</i>	.repeat

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

Index

A

ArchiveLibrary keyword 30, 53
audio file extensions 36
Audiodevice keyword 26

B

BackburnerManagerGroup keyword 39
BackburnerManagerGroupCapability 40
BackburnerManagerHostname
keyword 38
BackburnerManagerPort keyword 38
BackburnerManagerPriority keyword 39

C

CleanerDestinationPath keyword 41
ClipManagementDevice keyword 27
ClipMgtDevice File keyword 29
ClipMgtDevice HDCAM keyword 27
ClipMgtDevice Tape keyword 28
ClipMgtDevice VTR keyword 27
ColourSpace keyword 50
conventions, in users guide 2
customer support
contacting 3

D

DefaultPDFViewer keyword 37
DefaultWebBrowser keyword 37
DiskHealthNotify keyword 14
dlcfg command 7
documentation for this release 2
DviRamp keyword 17

E

Emulator keyword 25

environment directory keywords 53

F

FontDPSAFM keyword 33
FontDPSBase keyword 32
FontProxyHighString keyword 33–34
Framerate keyword 50

H

Hires keyword 51
HtmlLog keyword 32
HtmlTemplates keyword 31

I

image file extensions 35
installation
updating configuration file with
xdiff 10

L

LogDailyReports keyword 41

M

MaxLibrarySize keyword 29
MemoryApplication keyword 15
Menu keyword 31
MidiDevice keyword 20
Model keyword 31

N

NetworkPanelDisplay keyword 42

P

project configuration files
 modifying 47
 template files 44
 viewing 46

S

SetupArchiveTape keyword 30
software initialisation configuration
 file 6, 14
 keywords 14
 modifying 8
 samples 6
 software initialisation 6
 starting with custom 10
 updating with xdiff 10
 viewing 7
starting the software
 with a custom configuration file 10

support
 contacting 3

T

TabletDriver keyword 20
TextDefaultFont keyword 32
TextFileEncoding keyword 35
Timecode keyword 50

U

users guide
 conventions 2

V

Video keyword 16
VideoPreviewDevice keyword 18
VideoPreviewWindow keyword 52
VTR keyword 21