

Special Features Manual for Cinelerra 5.1

(Last update 06/04/2016)

Table of Contents

1. Introduction
 - 1.1 How to Build Cinelerra 5.1 from Developer's Git Repository
 - 1.1.1 Notable Options and Caveats
 - 1.1.2. How Builds Really Work and Even More Options
2. FFMPEG Early Probe Explanation
3. How to Create FFMPEG Options Files
 - 3.1 File Naming Conventions
 - 3.2 Option File format / Content
 - 3.3 Complete Options File Example
 - 3.4 Ffmpeg Common H.264 Rendering
4. Viewing & Modifying Ffmpeg Format Options inside Cinelerra
5. FFMPEG Items of Note
6. Crash Dumps for Analysis
7. Speed/Fade Automation Usage and Auto Gang
8. How to Create a Blu-ray or Standard DVD
 - 8.1 Output Terminal Messages from Creating DVD
 - 8.2 Helpful Hints for Problems/Debugging DVDs Creation
 - 8.3 Additional Features for Creating Blu-ray/standard DVDs
9. Recording Web Media or Television and Playback
 - 9.1 Record Web Media by real-time Audio Capture/Screen Capture
 - 9.2 Digital Video Broadcasting (DVB) Capture Configuration
 - 9.2.1 Overview
 - 9.2.2 Initial Setup for Digital Video Broadcast Capture
 - 9.2.3 Watching TV using Cinelerra
 - 9.2.4 Recording TV to Capture at some Future Time via Batch
 - 9.2.5 Recording Immediately to Capture a current TV program
 - 9.2.6 Watching a Previously Recorded TV program's
 - 9.2.7 Remote Control for DVB
 - 9.2.8 Android Remote Control for DVB
 - 9.2.9 How to Setup your Remote Device to work with Cinelerra
 - 9.3 The commercial DB... What is that? A TDB – Traveling Data Base
10. Themes User Interface
11. Resizable Fonts/Icons Support
12. Vicons – aka Video Icons
13. Toolset Visibility Support
14. Menu Bar Shell Commands
15. Last Play Position Memory
16. Audio H/W Gain Support
17. Valgrind Support Level
18. CFLAGS has -Wall
19. Multi-Viewer Window Support
20. Porter-Duff Operators and Overlay Modes
21. Titler Font Kerning

Appendix A – Crash Handling Tips

1. Introduction

The CV Cinelerra Manual covers “The Secrets of Cinelerra” and can be found at:

https://cinelerra-cv.org/docs/cinelerra_cv_manual_en.html

This manual covers features or areas of Cinelerra that are different or new in the 5.1 version from the standard CV Community version. Many of the new features were created by a lot of different people and were merged in from various sources and each person is to be commended for their efforts. The content of this manual also came from various people/places/email and thanks to all.

1.1 How to Build Cinelerra 5.1 from Developer's Git Repository

These are generic build instructions for building Cinelerra 5.1. Known to work on ubuntu, mint, suse, fedora, debian, and centos. It has not been tested on every single possible distro yet so you might expect to have to make some minor changes. It has failed miserably on FreeBSD. At least once, a Gentoo version was successfully created. A 32-bit Debian version has been built by wrapping the appropriate stuff around the ubuntu binary.

Alternatively, there are some pre-built 5.1 dynamic or static binaries which are updated on a fairly regular basis (as long as code changes have been made) available at:

<https://cinelerra-cv.org/five>

There are 2 kinds of builds, the default system-build and a single-user build. A system build has results which are installed to the system. The majority of the files are installed in the standard system paths, but some customization is possible. The single user build allows for running completely out of a local user directory so it doesn't affect the system.

We recommend the single-user version when possible. It makes it very easy to install a new version without having to delete the older version in case you want it for backup – once you are happy with the new version, all you have to do is delete the entire old directory path. Also, if you install a new Operating System version and if you have Cinelerra on separate disk space that is preserved, you won't have to reinstall Cinelerra. In addition for purposes of having the ability to interrupt or to see any possible error messages, if you start the application from a terminal window command line you will have more control to catch problems. The system builds can be useful in a university lab setting where there are possibly multiple users, or multiple versions.

There are two notable differences between “standard” views of Cinelerra and this implementation for the system builds. Both of these can be configured during installation. These differences make it possible to have several different versions installed without having them “walk” on each other.

- 1) application name can be set during installation and defaults to: “cin”
- 2) the home configuration directory can also be set and defaults to: “\$HOME/.bcast5”

To do a system build, you should read the README that is at the top level after you get the source.

- 1) You need at least 2.5GB of disk storage to operate a build. Need to have “git” installed.
- 2) Obviously in order to install into the system, you must run as root.
- 3) The "git" step has to download a bunch of stuff (approx 100MB) so allow time.
- 4) Run the following commands (this takes awhile):

```
cd /<build_path>/          # this is where you need the 2.5GB of disk space
git clone --depth 1 "git://git.cinelerra-cv.org/goodguy/cinelerra.git" cinelerra5
cd cinelerra5/cinelerra-5.1 # toplevel directory
```

NOTE: if your system has never had Cinelerra 5.1 installed, you will have to make sure you have all of the compilers and libraries necessary. So on the very first build you should run:

```
./bld_scripts/bld_prepare.sh <os> # where <os> represents the Operating System of
centos, fedora, suse, leap, ubuntu, debian.
./autogen.sh
./configure --prefix=/usr # optional parameters can be added here
make 2>&1 | tee log # make and log the build
```

- 5) Check for obvious build errors:

```
grep "\*\*\*.*error" -ai log
```

If this reports build errors, email the log which is listed below to cinelerra@lists.cinelerra-cv.org :
/<build_path>/cinelerra5/cinelerra-5.1/log

- 6) If there are no build errors, finally just run:
make install
- 7) If it all worked, you are all setup. Just click on the “blue lady” cinelerra desktop icon.

To do a single-user build, read the README that is at the top level after you get the source.

- 1) You need at least 2.5GB of disk storage to operate a build. Need to have “git” installed.
- 2) Recommend you build and run as root, just to avoid permission issues initially.
- 3) The "git" step has to download a bunch of stuff (approx 100MB) so allow time.
- 4) Run the following commands (this takes awhile):

```
cd /<build_path>/          # this is where you need the 2.5GB of disk space
git clone --depth 1 "git://git.cinelerra-cv.org/goodguy/cinelerra.git" cinelerra5
cd cinelerra5/cinelerra-5.1
```

NOTE: if your system has never had Cinelerra 5.1 installed, you will have to make sure all of the compilers and libraries necessary are installed. So on the very first build you should run as root:

```
./bld_scripts/bld_prepare.sh <os> # where <os> represents the Operating System of
centos, fedora, suse, leap, ubuntu, debian.
./autogen.sh
./configure --with-single-user # the “with-single-user” parameter makes it so
make 2>&1 | tee log # make and log build (check for errors before proceeding)
make install
```

Then just start the application by keying in: `./cin` in the bin subdirectory OR add a desktop icon by using the appropriate directory to copy the files to, run as root, and edit to correct the directory path.

```
cd /cinelerra_directory_path
cp -a image/cin.{svg,xpm} /usr/share/pixmaps/
cp -a image/cin.desktop /usr/share/applications/cin.desktop
change the "Exec=cin" line to be "Exec=<your_directory_path>/bin/cin"
```

The preceding directions for doing a single-user build has been meticulously followed to build and run on a newly installed ubuntu 15 system WITHOUT BEING ROOT except for the bld_prepare.sh and creating the desktop icon.

1.1.1 Notable Options and Caveats

These procedures and the 5.1 Cinelerra software have all been run as “root” on home laptops/desktops. This provides the best chance to ensure all works correctly and also allows for handling errors, other problems and potential crashes with the most success. Included in this section are some of the build variations easily available for normal builds.

To see the full list of features use: `./configure --help`

The default build is a system build which uses: `./configure --without-single-user`

In the single-user build, the target directory is always “cin”. Because this is also the developer build, constant names are used throughout. However, you can rename files after the install is complete.

If your distro/operating system has issues with the default install to /usr/local, you might have to change the location to /usr for a system build. Then you will have to use:

```
./configure --prefix=/usr
```

If you wish to change the default directory for a system build you will have to add the destination directory path on the “make install” line. For example:

```
make install DESTDIR=<your selected target directory path>
```

The application name can be set during installation, but defaults to cin so that the 5.1 build can coexist with other Cinelerra builds if necessary. To override the default cin name, use:

```
./configure --with-exec-name=cinelerra
```

The home configuration directory can also be set, but defaults to \$HOME/.bcast5. For example:

```
./configure --with-config-dir=/myusername/.bcast5
```

NOTE: when you specify parameters to the configure program, it will create a make file as a consequence. Since in a make file, the \$ is a special character, it must be escaped so in order to represent a \$ as part of an input parameter, it has to be stuttered. That is, you will need \$\$ (2 dollar signs) to represent a single dollar sign.

To build a 10-bit version for x264 and x265, it is a sad truth that the x264 and x265 libraries require “special” builds to encode 10/12 bit output, and so a separate version of cinelerra is needed to create these kinds of codec output. To build these:

```
./configure --with-exec-name=cinx --enable-x264_hidepth --enable-x265_hidepth
```

Unfortunately, these libraries can only be configured to encode either 8-bit or 10/12-bit but not both

compiled simultaneously. The 10-bit compiled Cinelerra can read 8-bit or 10-bit but can only render to 10-bit. Newer video equipment may take advantage of the 10-bit color depth for improved output quality.

1.1.2 How Builds Really Work and Even More Options

This section describes how builds really work if you want to know more about making changes or understanding the process probably only for a developer or system administrator.

Builds occur in 4 basic steps:

- 1) unpack/patch source code
- 2) configure build
- 3) make build targets
- 4) installation

So, an example of what happens in 4 steps for a single-user build would be as follows:

- 1) unpack/patch source code
`git clone --depth 1 "git:/.../target" cinelerra5`
`./autogen.sh`
- 2) configure build
`./configure --with-single-user`
- 3) make build targets
`make 2>&1 | tee log`
- 4) installation
`make install`

A lot of things can be tweaked to change the results. Mostly these changes are parameters to the configure step, which can change important build related items, like the application name, or where and what the target system directories should be. This makes it possible to have several versions at the same time on the same computer if needed. To see what it is that the makefiles use to build cinelerra, look at the resulting toplevel `global_config` file which is created by the `./configure` step.

Building cinelerra requires many thirdparty libraries, and it is recommended that you use the static build version included in the git repo. Some of them are patched, and fix significant bugs. It is important to note that because system installation historically has been with as much shared objects as possible, the defaults are that any system library detected during configuration setup will be used, when the package is built “`--without-single-user`”, which is the default build. To build with static thirdparty libraries for system install to the system `/usr` area, use:

```
./configure --enable-static-build --prefix=/usr
```

Sometimes, additional package parameters and variables are needed during thirdparty builds. These optional values occur before and after the “configure” and “make” commands during a build. A presentation of the format of the package qualified variable names and how they appear in the build procedure are:

| | |
|----------------|---|
| pkg.cfg_vars | prepended to configure |
| pkg.cfg_params | appended to configure |
| pkg.mak_vars | prepended to make |
| pkg.mak_params | appended to make |
| pkg.cflags | added as CFLAGS+=\$(cflags) to pkg.vars |
| pkg.cppflags | added as CPPFLAGS+=\$(cppflags) to pkg.vars |

These steps are done for EACH of the packages in the thirdparty build:

```
<pkg.cfg_vars> ./configure <pkg.cfg_params>
<pkg.mak_vars> make <pkg.mak_params>
```

The thirdparty Makefile has a set of default vars and params used to build each of the needed thirdparty packages, but you can specify new or overriding values for these Makefile substitutions. These thirdparty build config changes are specified in the toplevel file: cin_config. By using this file, you can save the configuration changes made for the current build to use the next time you do a new build. For example, to add an include file path to the giflib build, add this line to cin_config:

```
giflib.cflags := -I/usr/local/include/giflib5
```

To have a param/var change apply to all thirdparty builds, use:

```
CFG_VARS, CFG_PARAMS, MAK_VARS, MAK_PARAMS
```

CFLAGS, CXXFLAGS and LDFLAGS are forwarded to the thirdparty build environment via:

```
CFLAGS=-ggdb ./configure --with-single-user
```

However, there is no guarantee that the thirdparty build will honor the environmental flags.

Finally, there are build controls, which enable/disable and set build features.

A few of the more useful ./configure --parameters are:

```
--with-jobs=n      where n=number of build jobs; defaults to 1.5*cpus+2.
--enable-static-build  build all 3rd party libs; defaults to yes if single-user, else no.
--with-single-user    build installs to <build_path>/bin; no system installation.
```

The ./configure command builds “global_config”. The global_config is read by the thirdparty/Makefile to create the recipes and definitions used by the thirdparty build. To see the recipes and definitions, run these commands after you have run ./configure, so that global_config is available:

```
cd thirdparty
make rules
make list
```

2. FFMPEG Early Probe Explanation

When you open media, a series of libraries and codec functions are used to "probe" the data, to see if it

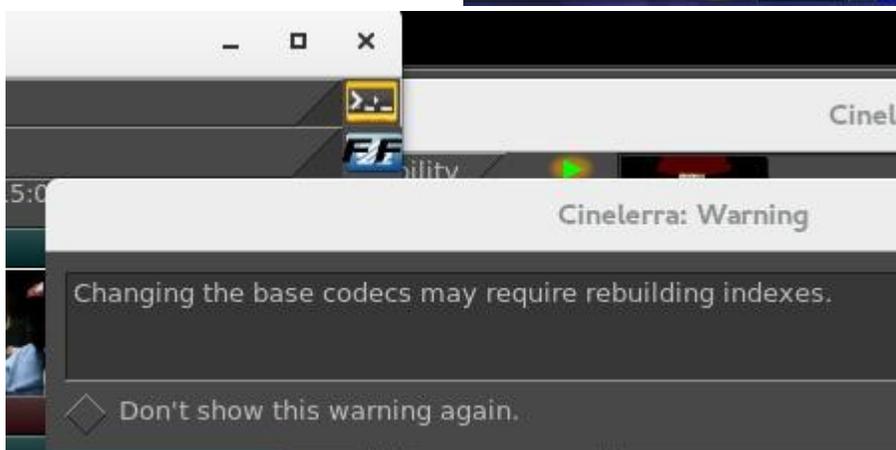
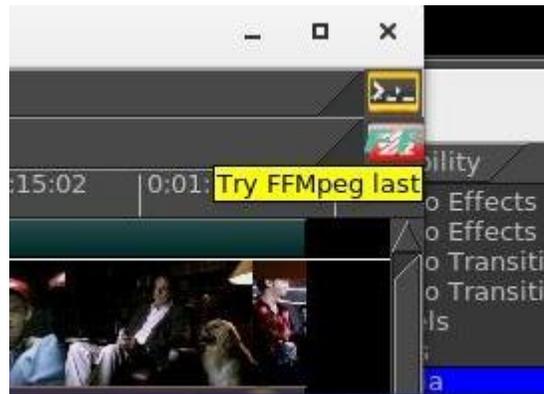
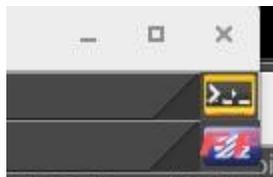
can guess the type of file format and codec parameters needed to correctly decode the file. If ffmpeg probes early - "Try FFMpeg first" is in effect for the button - it will usually find some way to try to decode just about any contemporary media file. But there are some times that the built in codecs are actually a better choice. A lot of this may fall into the category of personal preference. For example, some may prefer the mpeg library in the cinelerra code over the ffmpeg code because it has more decoding capability and seems to be more robust when the media is damaged. In that case you will want the *FF* button to read "Try FFMpeg last" in the upper right hand corner of the main window.

So, if ffmpeg probes early, you will never get to use the built in libraries, and if you want to skip over buggy old libraries, use ffmpeg early probe enabled so that the newest code will be tried first.

When the icon is red, ffmpeg probes early is enabled and you will see it reads "Try FFMpeg first" when moving over the *FF* button in the upper right hand corner of the screen. When the icon is black, ffmpeg probes early is disabled so that ffmpeg probes late and it reads "Try FFMpeg last". The initial default state of the icon is off, that is, ffmpeg probes late. This is the original code behavior before ffmpeg code was added. Suggestion is to leave it on except in a few special cases where it may be better to have early probes disabled. When you mouse over the main menu ff toggle button, the text displays ffmpeg's first/last probe position.

The ffmpeg early probe state is saved between sessions. It is important to note that the various file indexes may need to be rebuilt if you change which codec is being used to decode the file. There is a warning popup to remind you when you change the default ffmpeg early probe state (unless you have checked the box to no longer show you the warning).

Screenshots below show (1) reddish colored *FF* in upper right hand corner of main screen indicating that ffmpeg early probes is enabled; (2) "Try FFMpeg last" indicator message for ffmpeg early probes enabled (note that the color is different because you highlighted the icon); and (3) black colored *FF* indicates ffmpeg will be used last and you are changing the behavior so that Cinelerra warns you accordingly.



3. How to Create FFMPEG Options Files

This section describes how the FFMpeg options files work for decoding and encoding and goes into great detail. It will make more sense if you look at cinelerra's ffmpeg config directory and the cinelerra menus at the same time. It is meant to include everything necessary for complete understanding. You will be able to personalize your own options files without knowing all of the information included below if you know the basics. The word encoding is used interchangeably with the word rendering.

The possible combinations for ffmpeg options files are literally combinatorial – that is a lot (factorial!). The allowed media file format / codec choices are much more flexible than you might realize. When the ffmpeg design was initially added, some parameter files which describe the choices which the program uses had to be created. There are way too many to enumerate in the deliverable Cinelerra package. Some quite detailed information for how ffmpeg options work is given here and hopefully, enough basics for simple understanding. It may all seem complicated at first, but will become obvious quickly.

3.1 File naming convention

In Cinelerra's ffmpeg configuration directory you will see files as listed and described below. File type and extension names are the key for Cinelerra's use of ffmpeg. Basically the .opts file extension represents options; .dfl represents defaults; and all the rest are media types. For example one media type is quicktime so that *.qt file names would be the “quicktime” choices. In the file names below, “ext” refers to a set of files with file names matching the *.ext file extension. And “typ” refers to a type of format / codec combination used, i.e, the media type.

In the ffmpeg configuration directory there are a series of options files used when encoding or decoding audio or video. They are read in the order from top to bottom and only the files needed for the current operation are added to the active configuration.

```
ffmpeg/ffmpeg.opts = global ffmpeg options, always used
ffmpeg/decode.opts = global decoder options, used when opening existing files for decoding
ffmpeg/encode.opts = global encoder options, used when creating new files for encoding
ffmpeg/audio/audio.opts = audio encoder options, used when creating audio streams
ffmpeg/video/video.opts = video encoder options, used when creating video streams
ffmpeg/plugin.opts = parameters for ffmpeg filters as audio/video plugins
```

Decoder options:

Normally, only ffmpeg.opts and decode.opts are used when reading / decoding files, but may be specialized if a <path>/media.opts exists for a given <path>/media.ext file.

Encoder Options:

Within the audio /video subdirectories of the first level ffmpeg directory, the typ.ext files are for encoder (rendering) setups.

ffmpeg/audio = directory of audio encoder settings
audio.opts = options used by all audio encoders
typ1.ext, typ2.ext, ... = are all *.ext type choices for encoding audio
ext.dfl = contains the default selection used when ext is first selected

ffmpeg/video = directory of video encoder settings
video.opts = options used by all video encoders
typ1.ext, typ2.ext, ... = are all *.ext type choices for encoding video
ext.dfl = contains the default selection used when ext is first selected

ffmpeg/format = presets needed to initialize audio / video formats

3.2 Option File Format / Content

For the option files a specific format must be followed in creating the file content.

In typ.ext encoder parameter files, the first line is defined as:

```
muxer codec  
or muxer codec | bitstream filter [ bitstream filter options ]
```

where the | represents piping the codec data through the bitstream filter.

The rest of the lines in the file should look as follows:

```
# in column one is a comment  
id1 value1  
or id2 = value2
```

Only one equals sign is allowed and it is just for readability. There may be any number of id / value pair lines in a media definition, including zero. A typical line might be:

```
bitrate 4000000  
or bitrate = 5000000
```

There are 4 special id's recognized by cinelerra which cause special processing. They are:

```
duration      (overrides the probe duration when opening media for decoding)  
video_filter  (adds an video stream filter, eg. edge detect, ... at the stream level)  
audio_filter  (adds an audio stream filter, eg. echo, ... at the stream level)  
loglevel      (sets the library logging level, as quiet, panic, ... verbose, debug)
```

All other id's should be in the ffmpeg documentation, and correspond to the global, muxer, and codec option names and values used by ffmpeg. For example to set the aspect ratio to 4:3, use:

```
aspect 4:3
```

Below shows an example: "decode.opts" which is used when the ffmpeg decoder is initialized.

```
# apply at init decode  
loglevel=fatal  
formatprobesize=5000000  
scan_all_pmts=1  
threads=auto
```

The encoder options you see in the cinelerra menus depend on the files in these directories, NOT THE CODE. If you add files, you will get to use more variety.

In the "cinelerra" directory, which contains the ffmpeg configuration folder, there are the choices the program uses. When you open an ffmpeg format popup dialog, the listbox contains all of the codec

types which are identified by the file.ext extensions. Decoding has only a few options, since the ffmpeg file probes determine most of the options by looking at the media being opened, but encoding media requires a lot of setup. Below are some of the folders and files used to determine the configurations used by ffmpeg to decode and encode files.

These extensions create audio / video media classes:

```
dvd m2ts mkv mp3 mp4 mpeg qt pro
```

which become the choices in the render pulldown menu

So if you want to create a "mov" codec class, add two new files to the ffmpeg configuration directory:

```
audio/aud.mov and video/vid.mov
```

Now you will see this as what you can choose in the rendering choices for ffmpeg.

Inside the file you will see that the first line is special. It is the muxer and codec. For example:

```
h264 libx265
```

The contents may be something like: <path>/video/vid.mov =

```
mp4 libx265  
bitrate 4000000
```

This will code an mp4 formatted file using the lib264 codec encoder.

For audio and video together, **the mux format must agree** between the aud.mov and vid.mov files when they are to be used together. The stream muxer must be the same for the all the streams in the file being written.

for example: <path>/audio/aud.mov =

```
mp4 pcm_mulaw
```

This will create mp4 media using audio format pcm_mulaw coding.

Both the audio and the video are using mp4 mux format, and so there will be 2 streams:

- 1) x265 video
- 2) pcm_mulaw audio

When the menu popup is created, there may be many choices for that class type, so you may want defaults. That can be specified as:

```
audio/<class>.dfl and video/<class>.dfl  
so: audio/mov.dft =  
aud.mov  
and: video/mov.dft =  
vid.mov
```

The above will be the default choice when the menu opens.

When you see problems in using the new options files you have created and put into place, add the following line to ffmpeg/encoder.opts:

```
loglevel=verbose
```

sometimes that will be enough to see what has caused a failure, or even catch unexpected results.

Now there is an EXCEPTION to all of the above because of a conflict between ffmpeg and the x264 person making the detection of default ffmpeg settings terminate with an error. If you get this error, you must work around this termination by including parameters that don't match 5 or more of the normal expected values. So you just have to change a few parameters to avoid the probe detection. Here is an example where you will notice the x264opts line tweaking values to throw off the detection/error termination code.

```
<path>/ffmpegvideo/test.mp4 =  
  mp4 libx264  
  preset=slow  
  x264opts keyint=25:min-keyint=4:qpmin=3:qpmax=33:qp_step=4:merange=8  
  crf 20
```

Another EXCEPTION is for some forms of aac, which is the reason had to add bitstream filters. If you think that you are in this situation, use:

```
<path>/ffmpeg/audio/test.mp4 =  
  
  mp4 libfdk_aac | aac_adtstoasc
```

For more examples, look around the ffmpeg dir for examples which may be close to what you are trying to use, and see if the parameters look usable.

This is quite complicated, but that is because ffmpeg has a lot of parameters and history. Good results are not that hard to create. Initially you should mostly use the defaults. If you send any new options files to cinelerra@lists.cinelerra-cv.org, it may be given consideration to being added to the baseline for the possibility for future deliverables.

3.3 Complete Options File Example

For illustrative purposes, here is an example of the options files that need to be added for using the ffmpeg ProRes 422 format. This makes it possible to transcode to h264.mov with FFmpeg retaining 10-bit yuv422p from the source to the target output video.

```
Add the file named “./ffmpeg/audio/acc256k.pro” which contains the following lines:  
  mov libfdk_aac | acc_adtstoasc  
  strict -2  
  b 256000
```

(Note that in the example above, even though the bitrate is set here to 256000, it can be overridden by the render menu settings).

Add the file named “./ffmpeg/audio/pro.dfl” which contains the following lines:
acc256k.pro

Add the file named “./ffmpeg/video/med422p10.pro” which contains the following lines:
mov prores
preset medium
pix_fmt yuv422p10le

Add the file named “./ffmpeg/video/pro.dfl” which contains the following lines:
med422p10.pro

Then to use and to get 10 bit depth and preserve depth from decode to encode:

- 1) load media
- 2) use settings->format to set the frame rate, sample rate/channels, aspect ratio, color model = rgb_float or rgba_float if blending
- 3) press Shift-R and select FFMpeg format type “pro”
- 4) select target path
- 5) check ok, and watch for messages in the terminal window

3.4 Ffmpeg Common H.264 Rendering

Because H.264 is so widely used, the method in 5.1 Cinelerra is outlined below. These setup steps make it easy to just get started.

File->Render
File Format->FFMPEG + mp4
Video Wrench->Preset->h264.mp4 + bitrate: 6000000 (or whatever) +OK
Audio Wrench->Preset->h265.mp4 + bitrate: 224000 (or whatever) + OK
Set your target path in: Render=>Select a file to render to
Set your timeline in: Render=>Render range + click Project
Set your insertion strategy: Replace project (or whatever)
Press OK to start rendering.

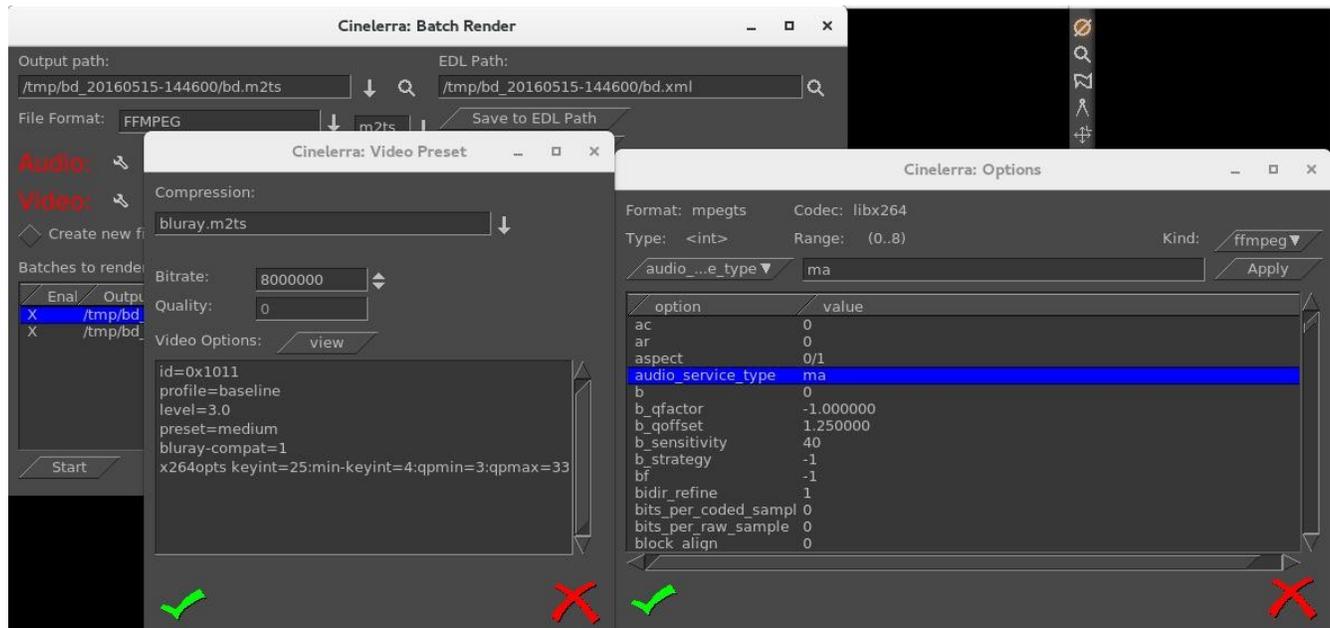
4. Viewing & Modifying Ffmpeg Format Options inside Cinelerra

There are thousands of options for using ffmpeg. Now it is possible to “view” the available options for a particular video and audio choice by using the “wrench icon” and then clicking on the “view” box. Ffmpeg has to be the selected file format for this feature to be visible. It makes it a lot easier since only the applicable options show up as opposed to everything that ffmpeg can do. These options are just “Hints” and some may be missing due to the way that ffmpeg options are coded – Cinelerra shows the option data ffmpeg has exposed.

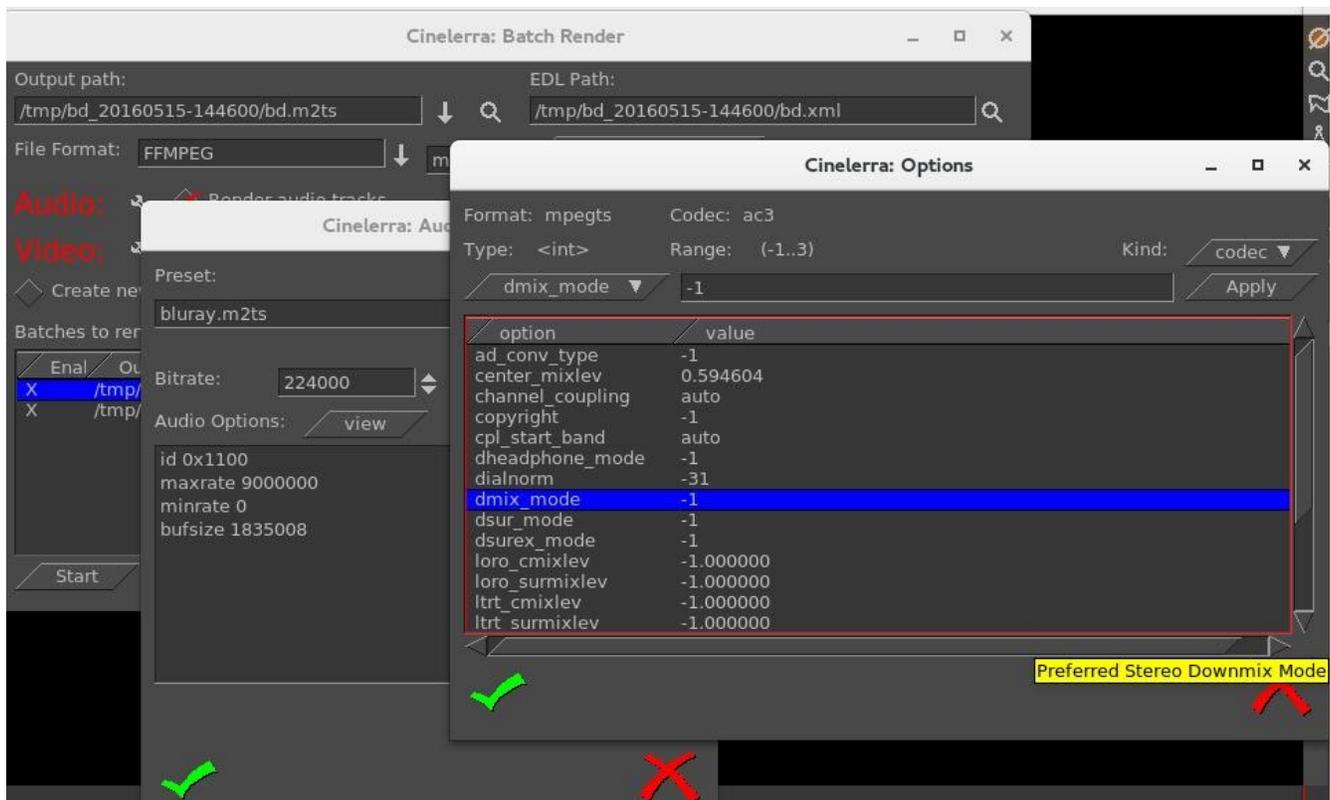
As an example, instead of reading the entire 264 library information, you only have to look at the shown available options. Both the video and the audio are browsable. The options visible in the Audio/Video Preset textbox are the final values which are used when rendering once you have checked OK. For assistance in choosing the options you want, use the view popup to see the objects that go with the selected format tool, highlight the option, modify the parameter value in the box at the top of the “Options” window based on what you want, and then click apply. Updates modify parameter values or new parameters will be appended at the bottom. Note that when you highlight an option a tooltip will show up when available in the lower right hand corner which describes the option. Also note that the Format and Codec types are shown on the top line of the Options window.

Parameters exist in 3 layers: ffmpeg, codec, and an interface layer. You can apply parameters to each layer. The top 2 layers are accessed with the Kind popup menu. The ffmpeg layer is topmost, and is selected as Kind: ffmpeg. It can specify many of the more common parameters, such as the bitrate, quality, and so on. The middle layer is selected as Kind: codec. These options can specialize your choices, and frequently includes presets and profiles useful for coding well known parameter sets, like profile=high422, preset=medium, or tune=film, etc. The interface layer may or may not be available. It is usually accessible only by an “opts” parameter, like x264opts=key=value:key=value:... These options are passed directly to the low level codec library.

The 2 screenshots next show “ffmpeg” for as the Kind: for Video first and then “codec” Audio second. Note the x264opts in the Video Preset window immediately below.



Note the yellow tooltip in the lower right hand corner describing the highlighted option. Also note the allowed “Range” values above the box provided for keyins.



5. FFMPEG Items of Note

Quality Option when rendering: Ffmpeg responds variably to the quality option in the render option but seems to respond well to bitrate. The subranges used by quality even seem to vary somewhat depending on how old the codec is. Some use 0 to 35, some use 0 to 500 or so. The quality is supposed to cause the codec to output data until the noise level is below a limit determined by the quality setting. Your specific results may vary.

Previous Changes when rendering: With ffmpeg there are 2 cases that the defaults will be used. The first time when you have nothing set up and any other time when you reset the render File Format in the Render Menu. Otherwise with ffmpeg if you change a video compression type for the render (for example h265.mp4 to h265.mp4), the settings will be from the previous session settings.

Outstanding Issues with ffmpeg: There are some problems that need to be addressed by the ffmpeg developer group that adversely affect Cineerra. These are stated below with the hopes that that group will fix them as time permits.

- 1) Make all the default parameters operational. When they are not, the Cineerra plugins can't be initialized since the initial state of the filter is not operational. If that is not possible, then provide a set of nominal parameters for each plugin, so that they can be used as the plugins initial default state.
- 2) Make the filter config function project the new parameter data into the filter function at any point during filter operation. This is so that continuous updates can be done as the plugin

operates.

3) Improve seek codec restarts. The past predictor must be reset or reconstructed after a seek. The only documented way to seek is open/seek/play. Reopening the format layer is very expensive.

6. Crash Dumps for Analysis

If you get a SEGV signal crash and can explain what steps you took, a /tmp/cinelerra_<pid>.dmp file is very useful for analysis. You can also use Ctrl-C in the controlling terminal window to force an INTR signal interrupt when you think the program is hung up. Email the /tmp/cinelerra_<pid>.dmp file along with setup information and input files that illustrate the problem to cinelerra@lists.cinelerra-cv.org .

- 1) Use the latest version of Cinelerra as you want to make sure problem hasn't been fixed.
- 2) Be sure gdb is installed (usually installed but if not, easily downloaded/installed by anyone)
- 3) Run as root (this enables using gdb to create a /tmp/cinelerra*.dmp).
- 4) Be sure Settings->Preferences->Performance->trap SEGV/INTR are checked.
- 5) Invoke the error. This should create /tmp/cinelerra-<pid>.dmp .

7. Speed / Fade Automation Usage and Auto Gang

Speed automation resamples the data at a higher or lower playback rate. The Fader affects the level of either the audio or video. Speed and fader automation can operate on all tracks of the same type, either audio or video, with a single click; or all tracks, both audio and video, with a double click. The curves to be affected must be both armed and ganged in the track patchbays to be included in the operation. You can see the advantage of having different media types ganged the same so that they will remain synchronized.

8. How to Create Blu-ray and Regular DVDs

This section describes how to create a blu-ray DVD or a regular/standard DVD. The settings and examples below more closely reflect creating a blu-ray but would be similar for regular DVD with, of course, the numbers being smaller. If referring to a blu-ray DVD, wording may be abbreviated to just "blu-ray". If referring to a regular/standard DVD, the wording may just be "DVD". Will use "DVDs" (plural) when mean either type. US stands for the United States and EU stands for the European continent. The DVDs created are unencrypted, unlike commercially available movie DVDs.

Requirements for creating DVDs is the hardware device to write the media on and, obviously, the blank media for either blu-ray or DVD. When generating DVD media, you will have to install dvdauthor (DVD) and udftools (blu-ray) if they are not on your system. Also, keep in mind that to mount filesystems for testing and to burn DVDs, you will have to be root since you have to have privileges, unless special permissions have been provided for a non-root user.

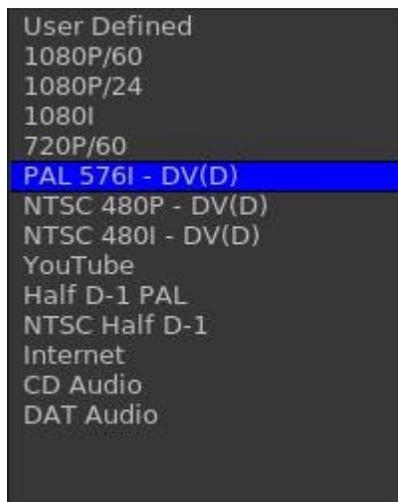
A big warning here – writing blu-ray DVDs and even regular DVDs can take a very large amount of clock time. Keep in mind that a blu-ray can contain 25 hours of viewing, so would take multiple hours to just write one.

The max disk space needed is a little over 100GB for 50GB Double Layer (DL) media or 50GB for a single layer blu-ray. A DVD of 4.7GB needs about 10GB disk space. You probably can get by with

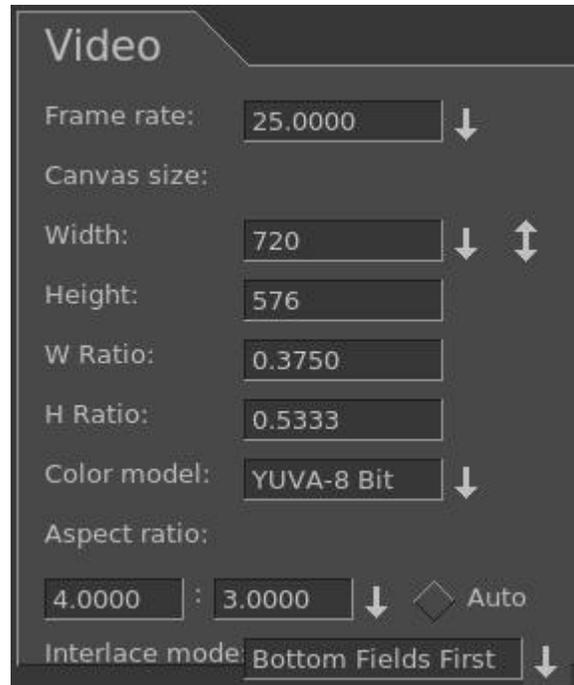
much less if the render for blu-ray is less than 25 hours of media. You will need twice as much disk space as the media holds to ensure you have sufficient space for working and copying.

Format settings shown in the Set Format window are set accordingly to an algorithm. Basically, you are "the boss" so it will take whatever you say in the asset format. It matches that against the known Presets available so that is what will be shown. If the asset format doesn't match any of the Presets default formats then that will be reflected as the "User Defined" format. In addition when you load media, the format will be set to "match" the source input and again if that matches a known preset, than that is what is shown. The PAL versus NTSC only comes into play when there is no known correct matching format when you attempt to create a DVD Render batch job. It is applied when you click OK. For example, if you load up a YouTube video, it will not match any known format and will choose PAL or NTSC based on time zone.

Screenshot of Presets in Set Format window



When choose PAL, values get changed in window to reflect PAL->



A quick set of **basic steps to create DVDs** is immediately below and usually **just using the defaults** will get you something. However there is a serious issue with interaction between the Operating System and bdwrite when creating a blu-ray that requires automount to be turned off. Refer to the details and more specific explanations below the following steps for how to do this.

- 1) Load your input source media via: File → Load files .
- 2) Choose PAL or NTSC for DVD or 1080P/24 for blu-ray in Settings → Format .
- 3) For blu-ray, choose BD Render or for PAL/NTSC, choose DVD Render in File → .
- 4) Designate a “work path” with sufficient disk space and then Chk-OK .
- 5) When the Batch Render window comes up, click on Start and the batch jobs will run.
- 6) Read the final messages echoed to the screen to see the command for burning... OR
- 7) Use the provided directory name to: `cd /<target directory>bd_(or dvd_)<date-time> .`
- 8) Load your media, format if needed, note device name to substitute for <bd> or <dvd>.
- 9) If rewritable blu-ray, use `dd if=./bd.udfs of=/dev/<bd> bs=2048000`
If write-once blu-ray, use `growisofs -dvd-compat -Z /dev/<bd>=./bd.udfs`
If any DVD media, use `growisofs -dvd-compat -Z /dev/<dvd> -dvd-video ./iso`

Any problems encountered will require that you read more information in this section to include specific details, helpful hints, and problem resolution.

Details and specific explanations to create blu-ray or regular DVD are provided here.

The general design of the DVD/blu-ray generation operations is to first render media using batch rendering and then terminate Cinelerra to start a script which creates the target device filesystem data. These scripts are the dvd.sh and bd.sh scripts written into the target directory. For DVD, the general plan is to write a directory <target>/iso with the dvd filesystem via dvdauthor and then generate an iso9660 filesystem and write it to a dvd via growisofs.

For blu-ray, the filesystem generation is slightly harder. First, it creates an empty filesystem image <target>/bd.udfs using mkudffs which makes a big hole for the filesystem data. The hole is made just a little bigger than the data written by bdwrite so that you don't have to write an entire 25GB or 50GB disc even if no data exists. This empty filesystem is loopback mounted to make it writable, and the linux kernel manages the filesystem image. The bdwrite program applies the blu-ray structure to the UDF filesystem by creating the needed BDMV blu-ray filesystem, which the kernel stores onto the image file bd.udfs. When udfs is unmounted, the kernel finalizes the disk image on bd.udfs. The bd.udfs image can be written directly to a blu-ray disk via dd or growisofs.

NOTE of IMPORTANCE: there is a serious situation with the interaction between the Operating System (OS) and bdwrite when creating blu-ray, that requires automount to be turned off. The blu-ray automatic script unmounts the blu-ray/UDF filesystem but the system has not finalized the directories so the OS creates a new loop file device and the data is loaded and cached for use by the new loop but it is stale. Consequences is that not all of the data is written where it should be. The solution is for the OS not to mount this second mount so we have to make sure it doesn't. There are 2 methods to fix this. The first and easiest is by using the following command to disable automount.

```
gsettings set org.gnome.desktop.media-handling automount false
```

This can be reversed when you have completed the blu-ray generation via:

```
gsettings set org.gnome.desktop.media-handling automount true
```

A different and more complicated method you can use to turn off automount is to download and install the dconf-editor. Automount is a system parameter and only needs to be done once unless you do not want automounts to always be disabled.

- 1) run: dconf-editor
- 2) select: org → gnome → desktop → applications->media-handling
- 3) uncheck: automount
- 4) close dconf-editor window

Immediately below are the detailed steps with explanations for creating a DVD or blu-ray disc.

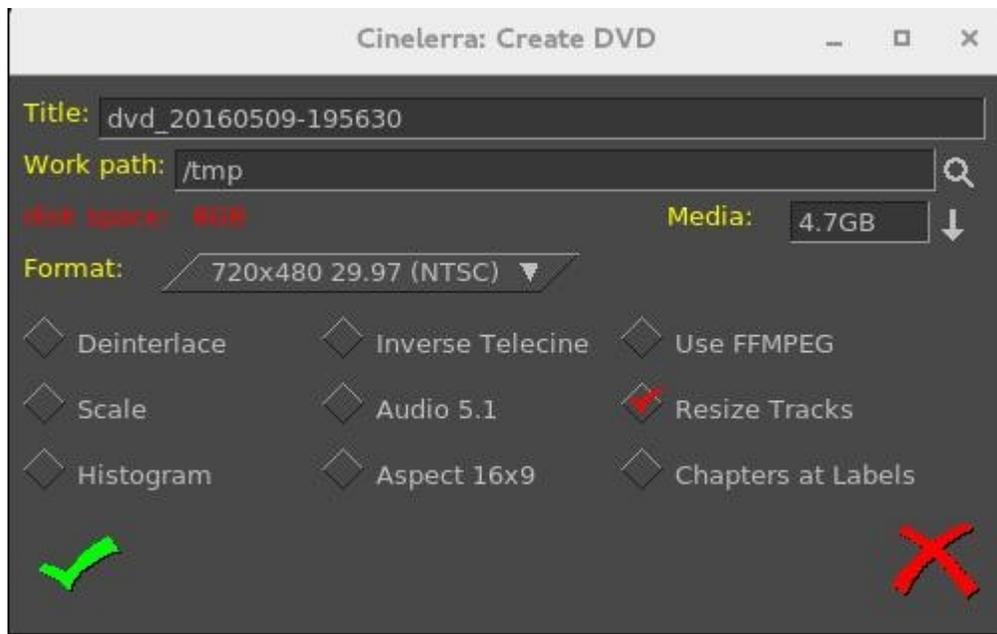
- 1) Construct a session with the desired presentation:

Format frame rates 23.97, 24, 25, 29.97, 50, 60 (for blu-ray, 23.97 is default and recommended; for DVD 29.97). Be sure to set the rendering parameters in the settings → format menu. Set the aspect ratio to suit presentation format dependent on your media (like 4:3, 16:9). Choose audio stereo or 5.1, again depending on your media. Target Geometry is 1920x1080 for most blu-ray and 720x480 (US) or 720x576 (EU) DVD.

- 2) From the main window, select file → BD Render or select file → DVD Render. Then:

Set the transcode Target Directory in the work path textbox !!! Another big warning here is to use a path that is NOT /tmp because a lot of linux distros wipe out /tmp on a regular basis via cron, crash, or reboot. Furthermore, they tend to use ram based filesystems, which will be way too small. You will lose your hard work so use disk space that is permanent. An icon in the form of a magnifying glass is next to the work path which shows you file names/paths. Disk space message will be green if you have at least 100GB/blu-ray or 10GB/DVD available space (the disk space message will be red to warn you if you have less than these amounts). Screenshots showing sufficient and insufficient disk space available is on the next page. Select desired features, check / un-check as appropriate. Click OK check button. It is **very important to realize** that when you check OK, the EDL will be saved and that will be used for batch job rendering. If you bring up the Batch Render and then change some parameters, they will not take effect UNLESS you remember to either check “Save to EDL Path” or “Use Current EDL” in the Batch Render window

Screenshots showing sufficient available disk space in green and then insufficient in red below.



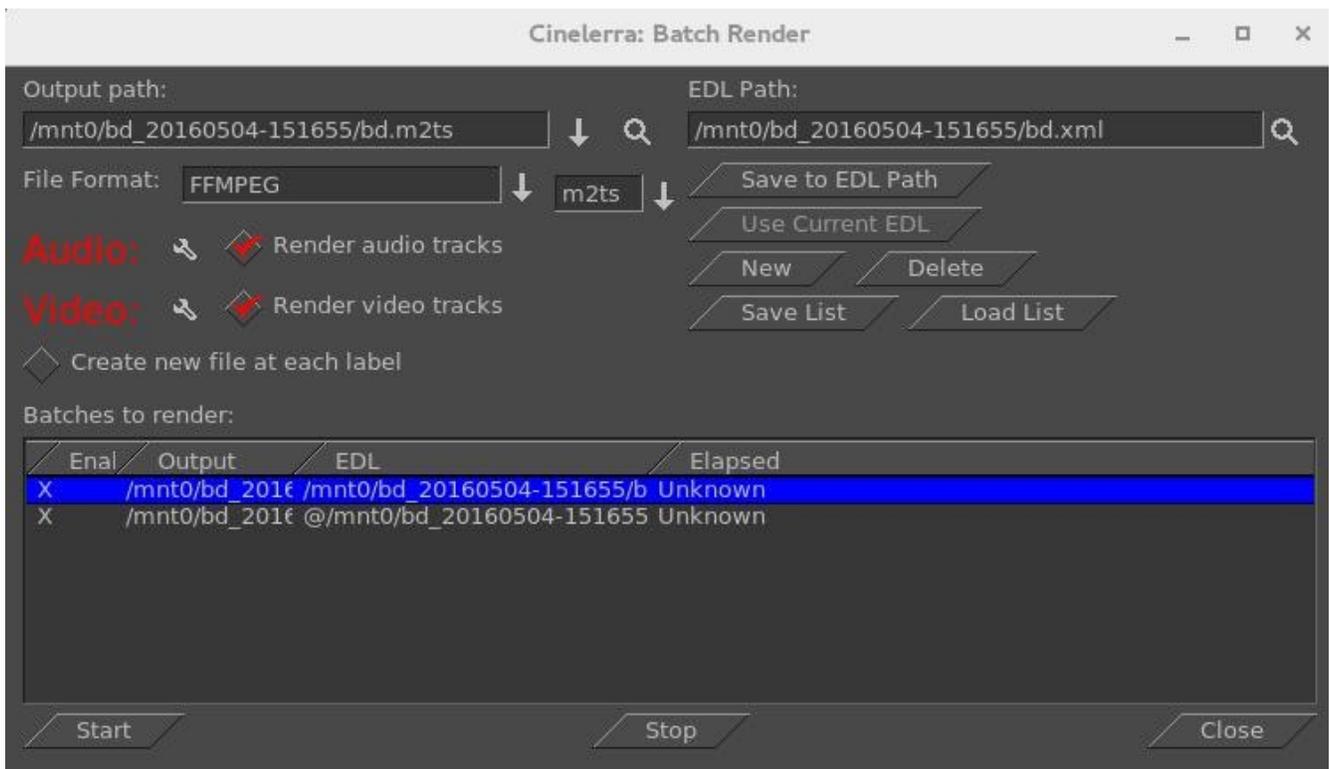
3) Batch render menu appears with “m2ts” format selected for blu-ray or “dvd” format selected for regular/standard DVD when File Format selected is ffmpeg in the previous “Create DVD” menu. It will work just fine without selecting ffmpeg for DVD and may be advantageous not to. Using the audio / video wrench tools (you will have to have the video batch job highlighted to manipulate the video or audio batch job for audio).

Setup the audio bitrate (192000 recommended). Data rate is 224K default.
Setup the video bitrate (6000000 - 12000000 recommended). 8Mb/sec is the current default.
Click OK check button.

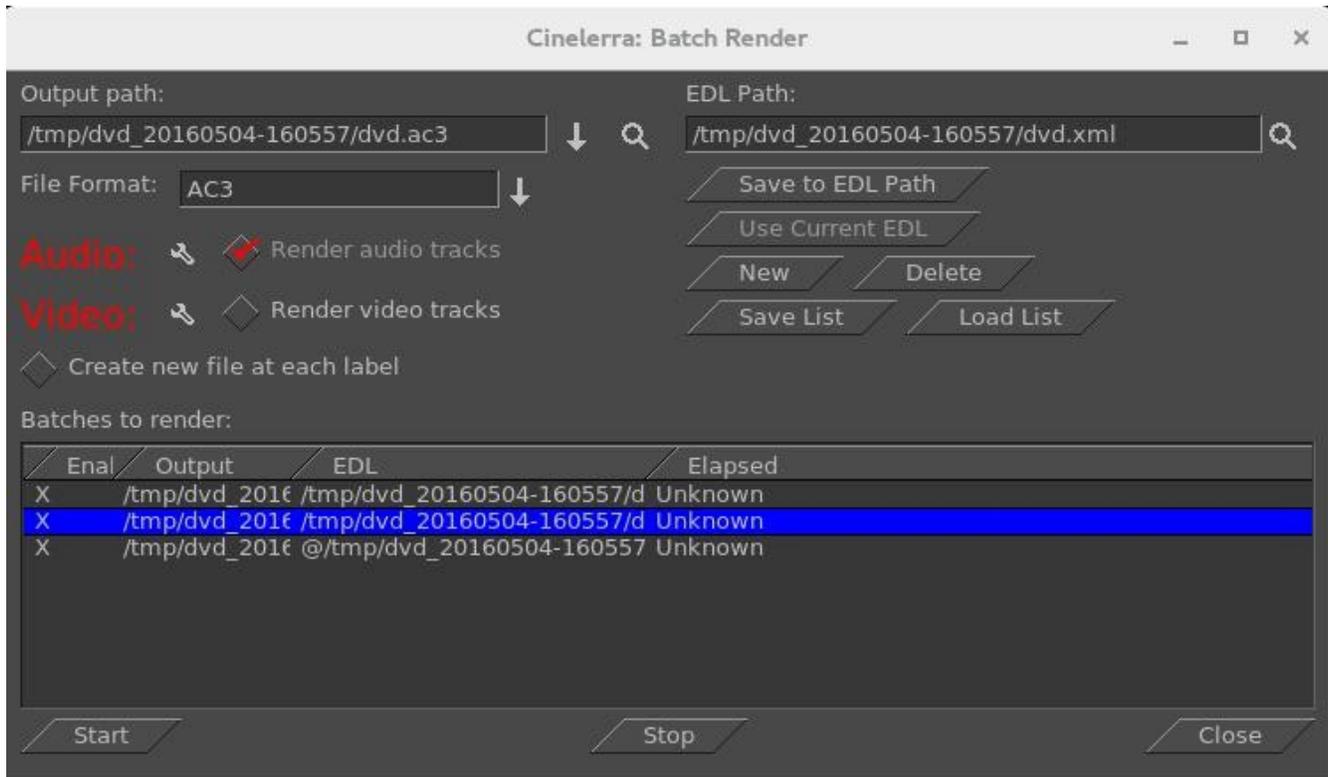
The default bitrate is the largest value possible. The actual 'target' bitrate is calculated based on a formula from the blu-ray/DVD code. It divides the media size (in bits) by the video time (in seconds) to find the bitrate that will "just fit" on the target media. This could create a weird bitrate if the media is large, and the video time is small, so the default/target bitrate is limited to 8Mb/s.

Batch jobs are then built and appended to the job list. You will see listed the batch jobs that are created to perform the rendering/tasks – 2 jobs for blu-ray, one for audio/video processing and one for the scripts. There will be 3 batch jobs created for DVD when not using ffmpeg to include one each for audio and video and then one more for the scripts. When you click on “start”, it fires off those jobs and you will see the rendering main window progress bar in the bottom corner reflecting the fact that it is currently processing. Be aware that the Cinelerra program will be totally shutdown AFTER the batch jobs finish and you will be at the command line prompt.

Screenshot below shows blu-ray creation with 2 batch jobs queued up and ready to go.



Screenshot below shows a DVD creation render ready to be started. Note that because it is not ffmpeg, the processing of video is done separately from audio. If need to modify the video tracks, which you can see is ghosted out, you need to highlight the first batch job listed. Be sure to highlight the first batch job before pressing Start so it runs all of the jobs and be sure the X for job enabled is set.



When you click on “start”, it fires off those jobs and you will see the rendering main window progress bar in the bottom corner reflecting the fact that it is currently processing. Be aware that the Cinelerra program will be totally shutdown AFTER the batch jobs finish and you will be at the command line prompt.

This will produce a new directory in your target path which contains a filesystem image file.

For example: /TargetDirectory/bd_20150820-093747

Directory and file names probably should not be changed at this time because the scripts and programs rely on the given names in order to proceed. You can change them later for your own purposes.

To test the file you just created use the command line interface; loopback mount the filesystem image which was generated in the target directory. For example if blu-ray:

```
cd /TargetDirectory/bd_20150820-093747/  
mount -o loop,ro ./bd.udfs ./udfs  
(check the media using a compatible media rendering tool like ffplay)  
umount ./udfs
```

To burn blu-ray media you will need to run from the command line interface. In the examples below, /dev/bd represents your blu-ray writer device (for example: /dev/sr1) and /dev/dvd represents your dvd writer device (for example: /dev/sr0).

Blu-ray Media

For rewritable blu-ray (recommended) (BD-RE):

Note - unwritten (virgin) media must be formatted first using:
dvd+rw-format /dev/bd (only done once and does not take very long)

To write or rewrite rewritable blu-ray media:

```
cd /TargetDirectory/bd_20150820-093747/  
dd if=./bd.udfs of=/dev/bd bs=2048000 (the growisofs command below also works)
```

To write blu-ray write-once media: (BD-R) (no pre-formatting needed):

```
cd /TargetDirectory/bd_20150820-093747/  
growisofs -dvd-compat -Z /dev/bd=./bd.udfs
```

DVD Media

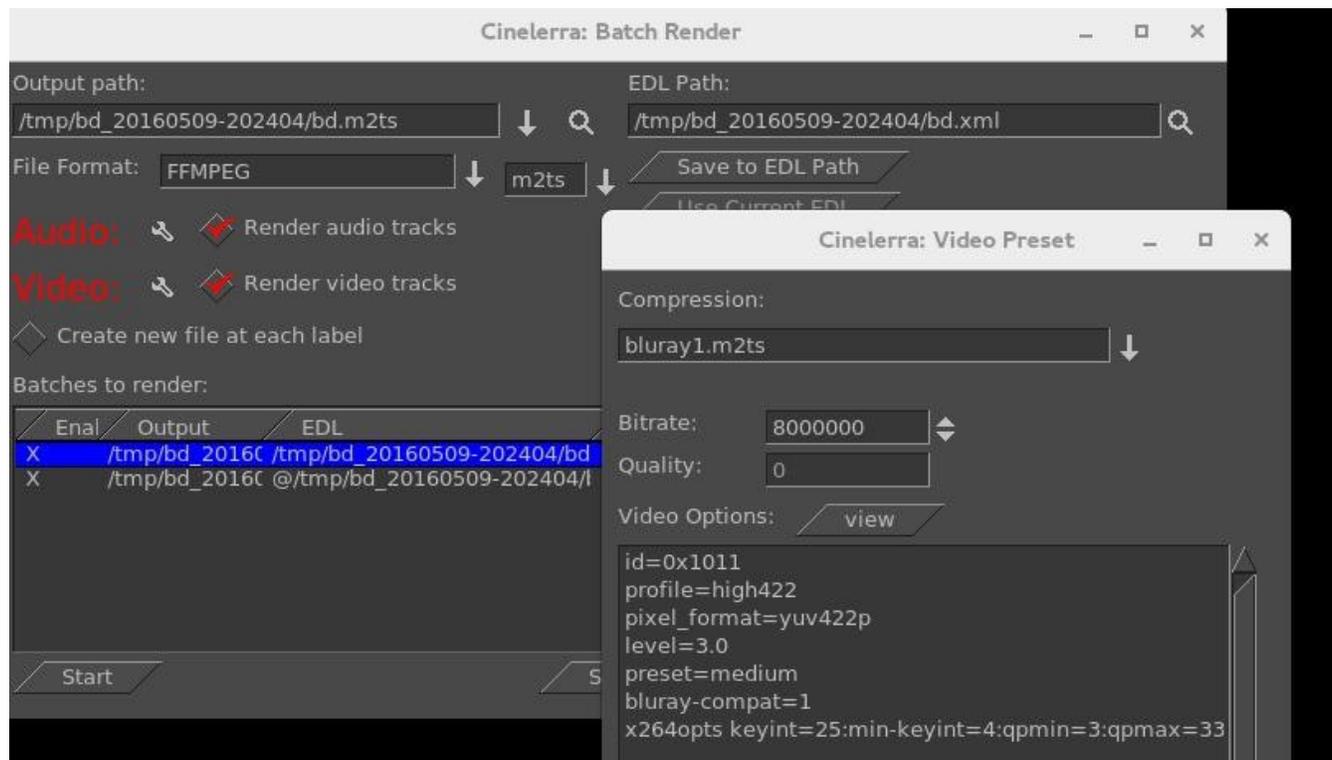
For rewritable DVD (DVD+RW):

Note – unwritten (virgin) media must be formatted first using:
dvd+rw-format /dev/dvd (only done once and does not take very long)

To write a DVD, load blank media and run the following from the command line (requires dvdauthor):

```
cd /TargetDirector/dvd_20160404-175416  
growisofs -dvd-compat -Z /dev/dvd -dvd-video ./iso
```

Screenshot showing the availability of 4:2 :2 for a Batch Render seen by clicking on wrench icon.



8.1 Output Terminal Messages from Creating DVD

Below is just an example of what the batch jobs generate and you will see on the terminal screen. It is just informational but will let you know if errors. In looking at any of the output, you can safely ignore the errors that read “Unsupported codec with id 100357 for input stream 0” - this comes from “nav-data” (navigation data). The example below is for DVD, not blu-ray which has very little logging.

```
FileMPEG::open_file: Running /mnt0/build5/cinelerra-5.1/bin//mpeg2enc -v 0 -v0 -b 8000 -q 1 -a 3 -F
4 -I 1 -M 4 -z t -f 8 -g 15 -G 15 -R 0 -o '/tmp/dvd_2016d.m2v'
FFMPEG::open_decoder: some stream times estimated
Render::render_single: Session finished.
FFMPEG::open_decoder: some stream times estimated
Render::render_single: Session finished.
running /tmp/dvd_20160407-113530/dvd.sh 1 /tmp/dvd_20160407-113530
  INFO: [mplex] mplex version 2.1.0 (2.2.7 $Date: 2012/11/17 01:55:16 $)
  INFO: [mplex] File /tmp/dvd_20160407-113530/dvd.m2v looks like an MPEG Video stream.
...
  INFO: [mplex] MUX STATUS: no under-runs detected.
DVDAuthor::dvdauthor, version 0.7.1.
Build options: gnugetopt graphicsmagick iconv freetype fribidi fontconfig
Send bug reports to <dvdauthor-users@lists.sourceforge.net>

INFO: default video format is NTSC
INFO: dvdauthor creating VTS
STAT: Picking VTS 01

STAT: Processing /tmp/dvd_20160407-113530/dvd.mpg...
STAT: VOB 32 at 15MB, 1 PGCs
INFO: Video pts = 0.133 .. 22.789
INFO: Audio[0] pts = 0.133 .. 22.789
STAT: VOB 46 at 21MB, 1 PGCs
CHAPTERS: VTS[1/1] 0.000
INFO: Generating VTS with the following video attributes:
INFO: MPEG version: mpeg2
INFO: TV standard: ntsc
INFO: Aspect ratio: 16:9
INFO: Resolution: 720x480
INFO: Audio ch 0 format: ac3/6ch, 48khz drc, 'en'

STAT: fixed 46 VOBs
INFO: dvdauthor creating table of contents
INFO: Scanning /tmp/dvd_20160407-113530/iso/VIDEO_TS/VTS_01_0.IFO
To burn dvd, load blank media and run:
growisofs -dvd-compat -Z /dev/dvd -dvd-video /tmp/dvd_20160407-113530/iso
```

8.2 Helpful Hints for Problems/Debugging DVDs Creation

This section contains helpful hints, how to initially check the results, and some information on determining what might have gone wrong and how to address it.

- 1) For first time users, taking the defaults seem to work very well.
- 2) You may want to use rewritable media to see how it goes before using permanent media.
- 3) Until you are familiar with the procedure, start with shorter input in order not to waste time.
- 4) Test the generated output with a compatible media rendering tool before burning DVDs.
- 5) Check the list of files and file sizes after the batch jobs are complete before burning DVDs.

For blu-ray creation, cd /workpath/bd_date-time directory and look for similar files:

bd.udfs bd.m2ts bd.sh bd.xml
udfs directory which is used as a loopback mount point

For DVD creation, cd /workpath/dvd_date-time directory and look for similar files:

dvd.mpg dvd.ac3 dvd.m2v dvd.sh dvd.xml
iso directory with VIDEO_TS and AUDIO_TS with non-zero size

- 6) The bd.sh and dvd.sh files are script files that you can carefully run manually from some start point to determine where the failure occurred. You must BE CAREFUL and know what you are doing and what directory you are in because dvd.sh contains an “rm” command and will delete files. The script takes a command line parameter of the directory where the file was rendered to and which is usually the directory where dvd.sh or bd.sh was created.
- 7) For blu-ray check to make sure you do not have any spurious loopback disks mounted that may interfere with the correct generation. Use the df command to check this and then the umount command to unmount these. Also, check to make sure you have used the gsettings command to disable automount.
- 8) For blu-ray loopback mount the <target>/bd.udfs image, and see if it has the BDMV filesystem written to it, and in particular a subdirectory named STREAM. Look at the results in ./udfs and check for the stream file which should exist in ./udfs/BDMV/STREAM/00000.m2ts and should have the same size as ./bd.m2ts.

```
mount -o loop <target>/bd.udfs <target>/udfs
ls -lR <target>/udfs
du -sc <target>/udfs
umount <target>/udfs
```

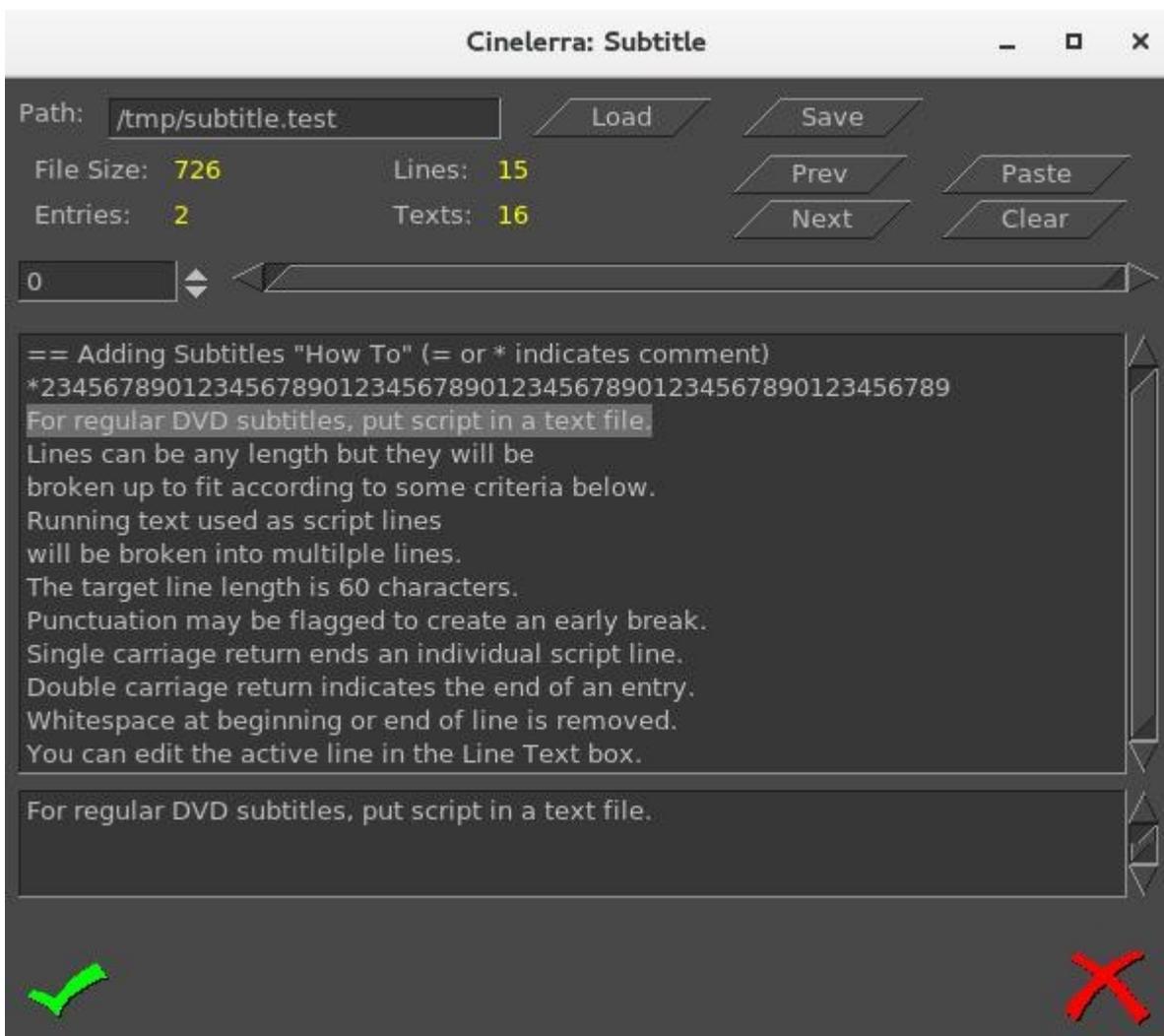
8.3 Additional Features for Blu-ray/DVD Creation

This section describes additional features of creating DVDs which can be very advantageous. It is quite easy to learn “how to” and rather simple, until it becomes tedious to do an entire 2 hour DVD!

Subtitles:

DVD (not blu-ray... yet) subtitles are added by using the main window pulldown File → Subtitle which brings up a window allowing you to type the filename of a previously generated text file containing the desired words/lines, the script. After entering the filename, click “Load” to read in your script. By creating a script file ahead of time, it lets you easily add dialog that was already written out and carefully edited for spelling and proper grammar.

Screenshot below shows the Subtitle window you will see.



The format of the script/text input file has specific requirements as listed below:

Lines can be any length but they will be broken up to fit according to some criteria below.

Running text used as script lines will be broken into multiple lines.

The target line length is 60 characters.

Punctuation may be flagged to create an early break.

Single carriage return ends an individual script line.

Double carriage return indicates the end of an entry and helps to keep track of where you are.

The “=” sign in column 1 indicates a comment seen in the script text to assist you in location.

An “*” at the beginning of the line is a comment and not a script line.

Whitespace at either the beginning of a script line or the end will be removed.

To put the subtitles onto your media, first add a subtitle track via the pulldown Track → Add Subttl . In the Subtitle window, note that there are 2 major textboxes. There is the “Script Text” textbox showing the current entry of text from your input file and there is the “Line Text” textbox showing the currently active text. In your subtitle track, select a timeline region (in/out or drag select with hairline cursor/highlight) to indicate the region where you want the active Line Text to be pasted. Then click the “Paste” button in the Subtitle window to paste the line onto the subtitle track. Silence will be added to the subtitle track in the places in the media where there are gaps.

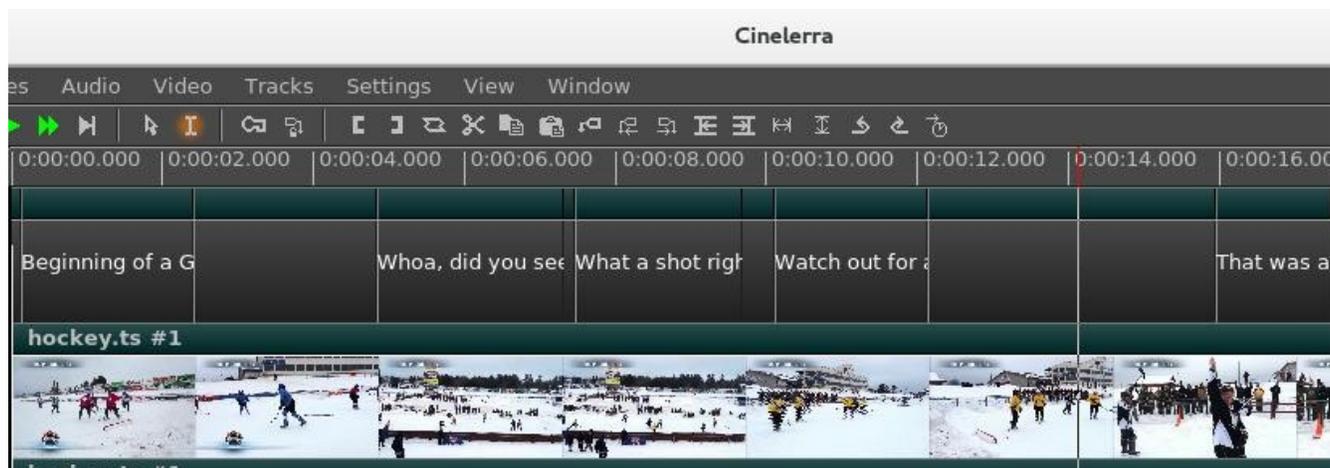
Editing in the Line Text box can be used to change the active script line. By double clicking the timeline over the subtitle track, you can reselect the active script line. The subtitle text will be reloaded into the Line Text box and can be edited and repasted as the new active subtitle text. You can also highlight multiple lines in the Script Text box and paste them (using the usual window paste methodology) into the Line Text box. After pasting to the timeline, the Line Text box will be updated with the next script line. If you click OK via the green check box before saving, the subtitle script position is saved with the session. This is convenient for continuing later where you left off.

To reposition the script, use the slider or tumbler buttons:

Slider bar to move through the text entries quickly.

Prev or Next buttons to go to the previous or next script line.

Screenshot below shows what the pasted subtitle script looks like in a portion of the main window.



9. Recording: Web Media / Screen Capture / Television and Playback

The recording application really only does one thing, capture media from some external source, and write it onto a file in a specified way.

| | | |
|-------------|------------------------|------------------------------|
| Path: | output media file path | |
| Start time: | weekday / time of day | to begin capture |
| Duration: | hrs:mins:secs | until end of capture |
| Source: | channel/device input | selection of input subsource |
| Mode: | timed | use start time / duration |
| | untimed | use transport controls |

The media file will be written using the format and codec specified in the settings → preferences → record tab. Only ffmpeg can record both audio and video simultaneously, and some ffmpeg formats require too much cpu to do a realtime compression. A setting which is more likely to be usable (requires less cpu/memory for realtime encoding) is ffmpeg / qt / mp4.qt with bitrates like audio/256000 and video/6000000.

9.1 Record Web Media by real-time Audio Capture/Screen Capture

Below describes the necessary steps for recording freely available media from the internet for your own personal use. You have to be on a system using pulseaudio, such as ubuntu, fedora, centos.

- 1) Start cinelerra and select Settings->Preferences->Recording
From a terminal (with a wide text window) run: `pactl list`.
You will see all of the audio sources and sinks on your system.
Identify the source associated with the normal output your system uses; example: Source #1.
Locate the source which monitors your normal audio output. For example: front stereo.
- 2) Choose a File Format and File Type (for example: FFMPEG & qt).
Check Record audio tracks.
Click Audio wrench tool:
Select mp4, for example.
Set Bitrate (256000 is a reasonable number).
Check Record video tracks.
Click Video wrench tool:
Select mp4 video, for example.
Set Bitrate 1000000 - 4000000 (bigger=better, but more cpu)
Select Audio In->Record Driver: ALSA (or whatever you have instead)
Set the Bits: 16 Bit Linear (probably, because reasonable and fast)
Select Device: the source from above pactl list search, for example Source #1)
Samples read from device: aprox dev buffer size (2k-16k probably)
Samples to write to disk: 131072 (a good size)
Sample rate for recording: 44100 (will automatically change)
Channels to record 2 (probably)
Uncheck Map 5.1->2
Gain 1.0

Select Video In->Record Driver: Screencapture

Set the Display: leave blank (probably) or use ":0.0" (default screen/display)

Frames to record to disk at a time: 30 (a good number)

Frames to buffer in device: (2-6) (probably)

Positioning: Software Timing (important)

Uncheck: Sync drives automatically

Size of captured frame: 720x480 (this is your choice, actual capture size; 600x320 youtube)

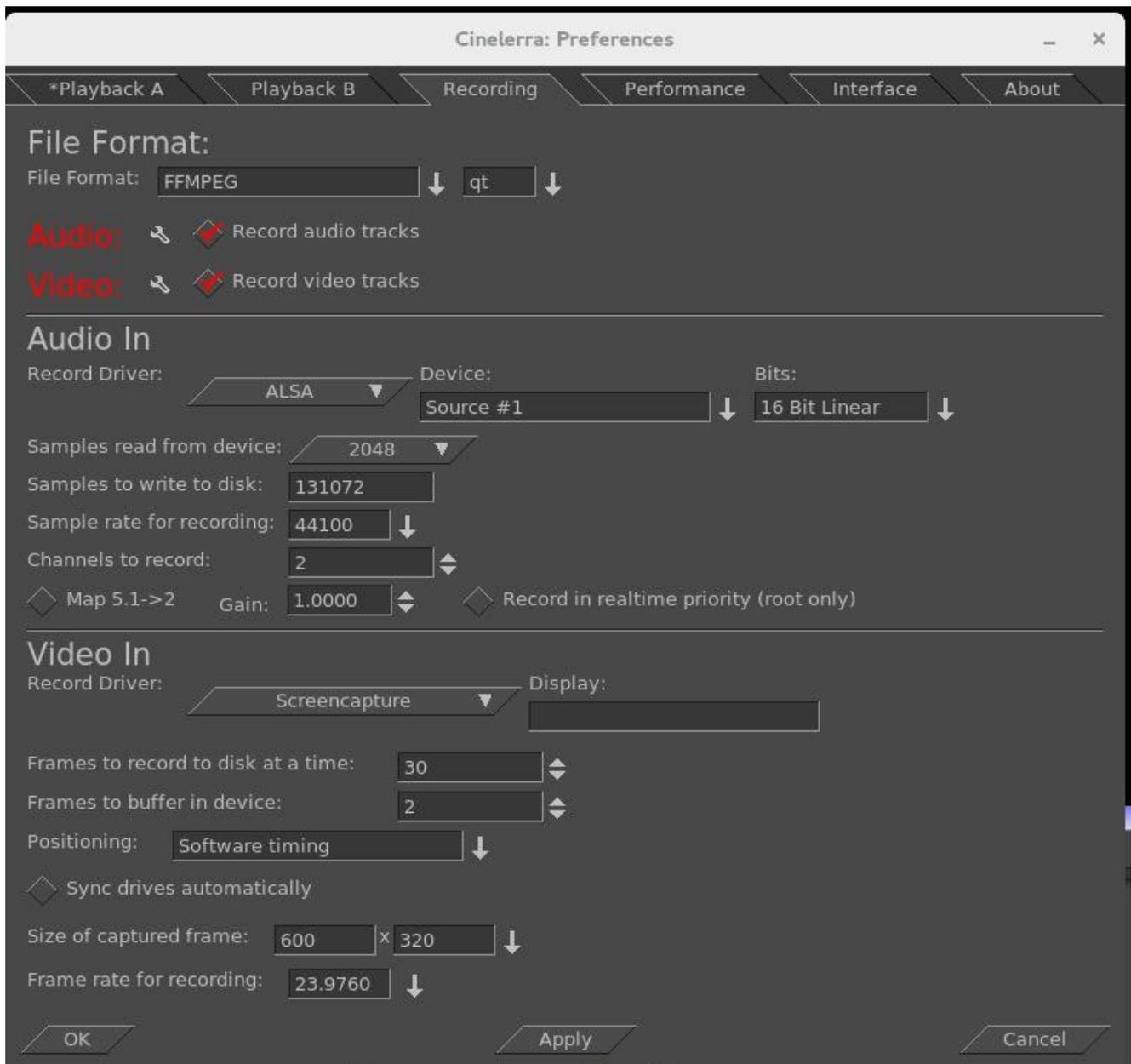
This defines a "screen capture" rectangular area on the display.

Try to make it the size you need to cover the screen playback area.

Frame rate for recording 23.97 (a good choice, not all choices work)

OK

Screenshot to illustrate some appropriate settings which are described above/below the image.



- 3) Make sure the Compositor window is up since you will want to see the captured media later. From the main window press (lower case r) (move windows and resize if needed) Using the mouse pointer, mouse over the "Video In" Recording monitor display. Press and hold the left mouse button and drag the display until the desired target screen capture area is properly positioned inside the record monitor "Video In" display. At first this may seem confusing, but keep in mind that what you are actually doing is positioning the portion of your monitor display you want to capture over the "Video In" window.
- 4) From the "Recording window" select a Path file name for the recording.
- 5) Select mode untimed.
- 6) Make sure the "monitor audio" is unchecked else audio becomes looped and is very loud.
- 7) Press Transport "start recording" (red circle button).
- 8) Operate the playback of the media source, i.e. start playing the web video.
- 9) When the media ends, stop on the playback application.
- 10) Press Transport "stop recording" button (white square button).
- 11) Before clicking chk-Ok, be sure to set the Insertion Strategy.
- 12) Press chk-Ok to review the capture, or X to cancel / dismiss application.

Screenshot of what the "Recording" display will look like.



Two things are happening during recording, the data is being written, and it is being rendered. When recording, if video and audio are not in sync, it will usually be video behind audio as opposed to the other way around because video is more difficult to decode. During recording, video frame and audio sample positions are checked and efforts to maintain synchronization are used. Below is an explanation for some of the less obvious settings for recording as shown in the previous screenshot.

Frames dropped – more input than cpu(s) can process. This is the number of frames dropped when the system is too slow to keep up with the video. This value serves as a warning that it is behind and is dropping frames to keep up. You might want to consider stopping to make some parameter adjustments and start over to reduce drops.

Frames behind – active input buffers. This is the number of frames that the video is behind and it will drop frames to get caught up. “Frames behind” depends on the synchronization source, for example, time as the synchronization. This means that frames are not being written as fast as they are being captured and can occur when the system is too slow to keep up. This situation can happen with older systems or cpus where there is more input than can be processed in a timely manner. So that when there are “Frames behind”, frames will be dropped.

The algorithm for determining how many frames to drop is as follows:

If the number of frames behind > 3 , only 3 frames will be dropped before it will then do 1 frame;
if the number of frames behind is 2, only drop 1 frame;
if only 1 frame behind, no frames will be dropped.

Drop overrun frames – checkbox when enabled, frames will be dropped. Drops occur when you capture a frame and there is no storage space available to save it, usually because buffer space is exhausted. This situation can occur when there are too many frames coming in too fast and the I/O system is not keeping up in emptying the buffers.

Fill underrun frames – checkbox if enabled, duplicate last frame when no input frame ready for output. As the tooltip states “write extra frames when behind”. This is desirable in order to keep audio and video synched with regards to sample and frame rates. Use underrun padding to fill frames when the input capture rate is lower than the output frame rate. Since demand is constant, you have to output something, so the last frame is duplicated as many times as needed to keep up. You should stop and adjust the frame/sample rates to match the input if not filling underruns, then start over. This situation can occur if the input stops for some reason like lightning interrupts the signal or the internet connection is too slow.

Transport controls - these control buttons mimic the functions of tape recorders from the old days.

Reverse button/left arrow – remnant from the past; does the same as the Start button really.

Red round button – this is the Start button to start recording.

White square – this is the Stop button to stop recording.

Red round button with white line through the center – start recording in single frame mode. The way this works is to just “take a picture now, take a picture now...”. The reason to use this mode is to get a still shot like you would with a camera. Sometimes input is continuous, for example looking at stars with a telescope -- surveillance goes on for hours, but you just want to take a picture now when something of significance interest comes up

Cron – The batch recording watcher, cron, is either Idle or Active. When you start or stop batch recording at specific times, there is a cron thread watching timers to perform the timed action. Idle/Active indicates whether the timers are running.

Position – this is a timebase which tracks frames/samples when obeying frame/sample rate. When a recording starts, it resets to 0. Timing is against audio (when available). Audio time and video time are based on position.

You can select synchronization time source “Positioning” in Settings → Preferences under the “Recording” tab. Possible choices for time base are:

Presentation Timestamps – use time code which is in both the audio and video media input stream. Uses these timestamps to sync the 2 streams.

Device Position – this is the device hardware position of where you are. It is usually only on the audio side.

Sample Position - Sample # or frame # divided by frame rate tells you where you are.

Software Timing – usually used for things like YouTube; it just will “take a picture now” ... “take a picture now” ... over and over again until you tell it to stop.

Positioning & Timing needs more detailed explanation for complete understanding and application. The overall goal is to maintain media stream timeline synchronization. The reason for providing different “Positioning” options is that different input media may have different timebase standards. Additionally, the input may be damaged. Damaged data can skew the timeline during presentation. The timebase standards make it possible to correctly resynchronize the media presentation to the original time position. For example, the transport layer may have timestamps provided in it. These timestamps record “audio time” and “video time” and are called “presentation timestamps”.

The kind of positioning used depends on the input device and the media format. In the transport stream example, the media has already been coded at least once as a stream and contains position information. In other situations, sample position or device position can provide timeline position. If the input device provides data at a variable or asynchronous rate, then software timers can provide timeline position. Sample position is position from the perspective of software, and device position is position from the perspective of hardware. The difference is buffer time. Device position is usually more accurate, but may not be available, or may contain errors.

Examples of media which usually has a particular type of positioning is:

- 1) mpeg has timestamps therefore Presentation Timestamps is a good recording choice;
- 2) raw media streams/screen capture have no timestamps, so tell it time via Software;
- 3) a web cam may supply data at variable rates, so again you would use “Software timing”;
- 4) an example of Device Position is audio (timestamps can come off the device);
- 5) if Software Positioning working for you, try Sample Position for really bad audio.

Other “Recording” settings are more pertinent to capturing and editing broadcast television with all of its many commercials. Their usage will be explained in more detail in a later section.

Label – create a label. During capture, put a time marker at the designated spot. When the recording gets pulled in later for review, you will see a green arrow marking the spot that was clicked.

ClrLbIs – clears all of the previously set labels.

Check for ads - check for commercials. In real time, labels are added when a commercial that matches a previous ad is already in the database (the TDB, Traveling Data Base).

9.2 Digital Video Broadcasting (DVB) Capture Configuration

You can do real-time capture of a full transport stream of Broadcast TV. Record television programs using Cinelerra by setting up ahead of time to start recording later at the specified time. Then you can use Cinelerra to watch later and easily fast forward through the commercials or edit the program, deleting the commercials, to watch uninterrupted. Some details may be slightly different than what it is in the United States. There are 3 requirements you must have to take advantage of this capability.

- 1) You need to have a suitable DVB Adapter such as the Hauppauge WinTV-HVR 950Q usb adapter. Make sure it is the linux version, not the new version.
- 2) Your DVB adapter on your PC should connect to a broadcast antenna input.
- 3) For now, you may have to run as the “root” user.

9.2.1 Overview

There is a lot of detailed explanation below to be thorough, but an overview is presented here.

- 1) Set up Record Preferences.
- 2) Scan the Channels.
- 3) Watch TV by selecting the desired channel.
- 4) Or set up a batch job to record a program and watch later via Shift-S in the main window.
- 5) Or record a currently running program by using the buttons in the Record window.
- 6) Load a previously recorded program.

9.2.2 Initial Setup for Digital Video Broadcast Capture

Listed below are the steps for the initial DVB setup. Setup will be preserved until you make changes.

- 1) Start cinelerra, and click Settings->Preferences->Recording
 - Select File Format checkbox: Realtime TOC (Table of Contents)
 - It is not necessary to set File Format at this point as it will automatically be set later.
 - Check Record Audio Tracks.
 - Check Record Video Tracks.
 - Select Audio In->Record Driver: DVB
 - Set the adapter path = /dev/dvb/adapter0 (probably)
 - Set the adapter dev = 0 (probably)
 - Set the Bits: 16 Bit Linear (probably)
 - Check: Follow audio config
 - Samples read from device: 2048 or approximate dev buffer size (2k-16k probably)
 - Samples to write to disk 131072 (a good size)
 - Sample rate for recording: 48000 (will automatically change, probably to 44100)
 - Channels to record: 2
 - Uncheck Map 5.1->2
 - Gain: 1.0
 - Uncheck: Record in real-time priority
 - Select Video In->Record Driver: DVB (note, File Format above changes to MPEG Stream)
 - Set the adapter path = /dev/dvb/adapter0 (probably)
 - Set the adapter dev = 0 (probably)

Check: Follow video config

Frames to record to disk at a time: 30 (a good number)

Frames to buffer in device: 2 (2-6 probably)

Positioning: Presentation Timestamps (**this is important and may not be already set**)

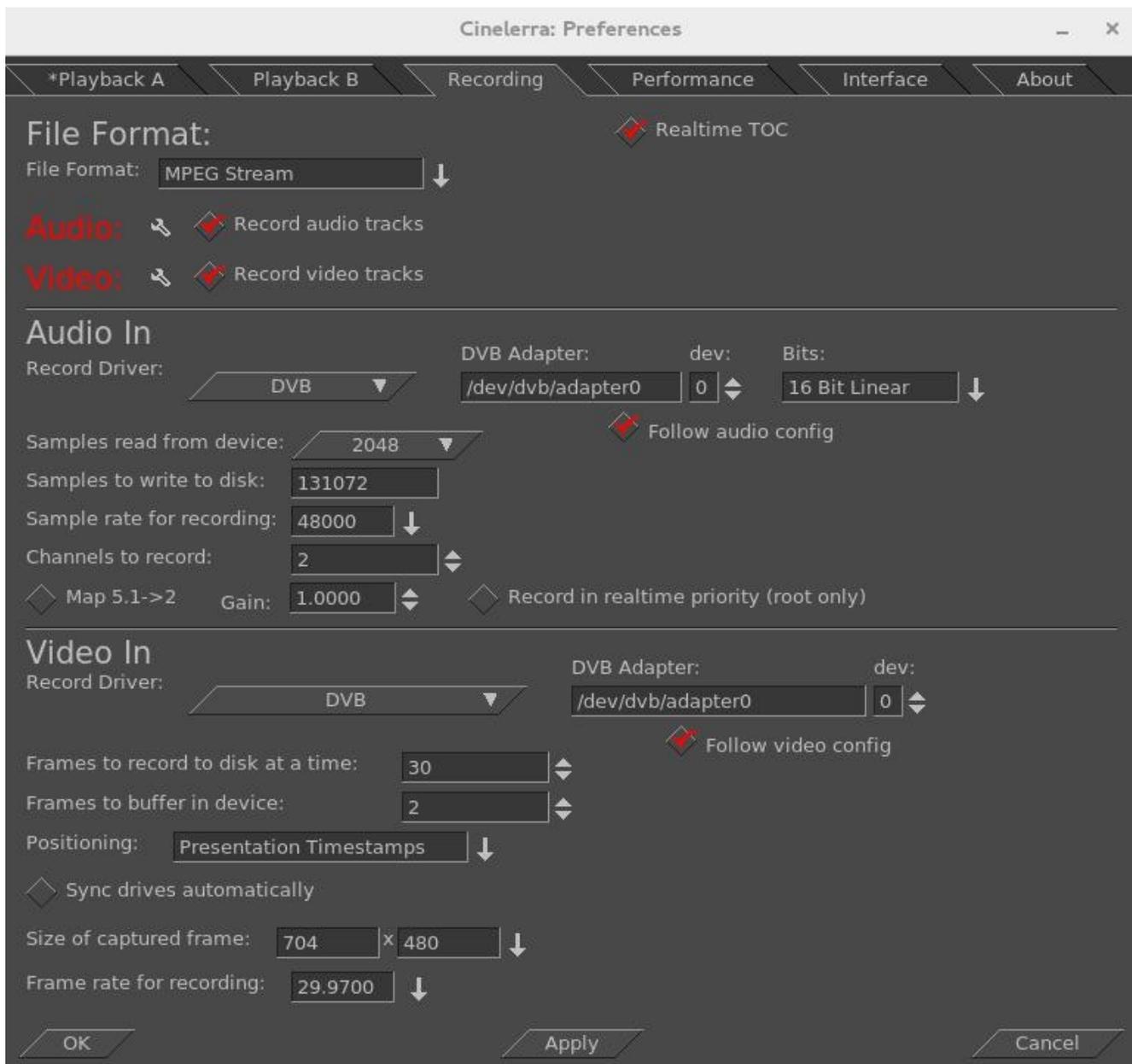
Uncheck: Sync drives automatically

Size of captured frame 720x480 (will automatically change)

Framerate for recording 29.97 (will automatically change)

Click OK

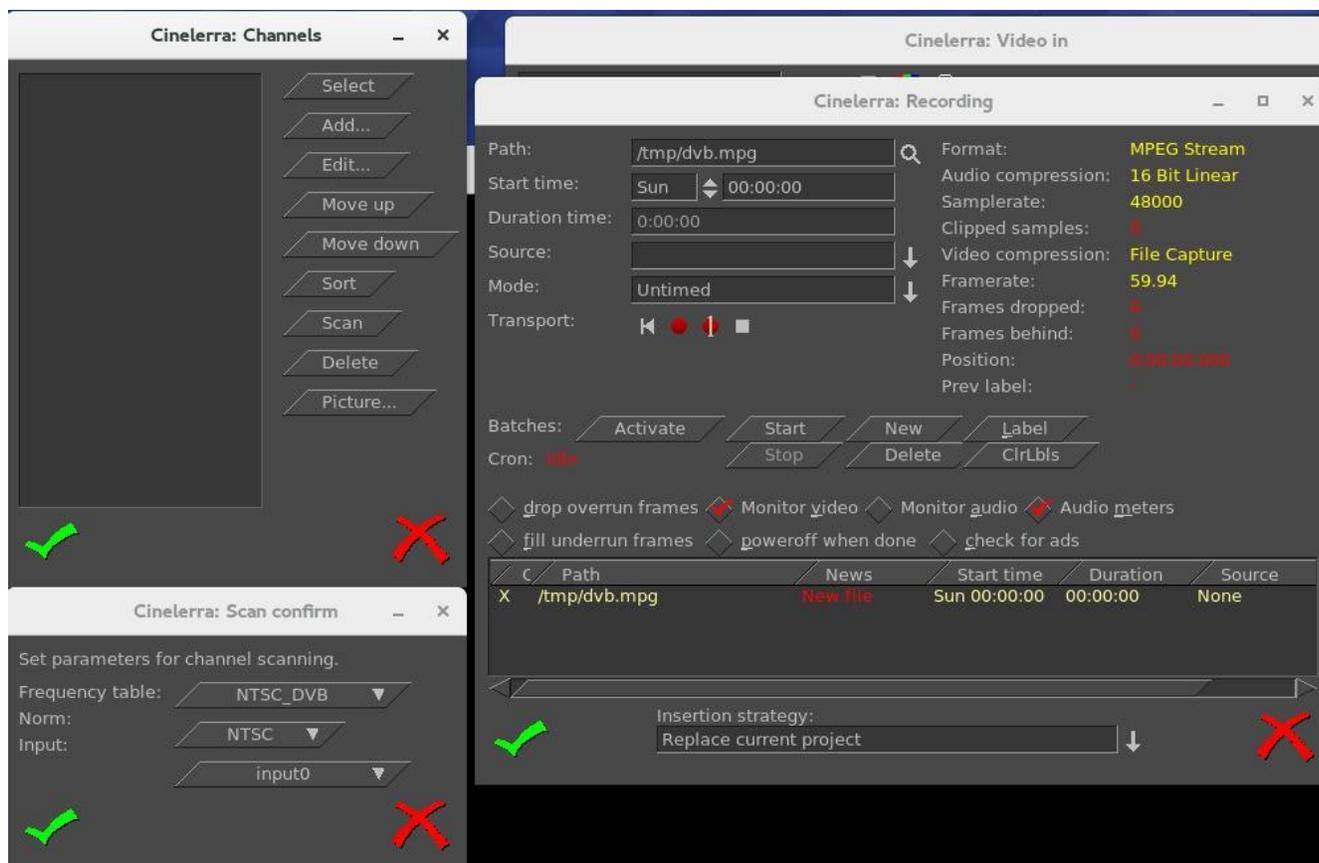
Screenshot below shows the Preferences window as a good example of what you should see.



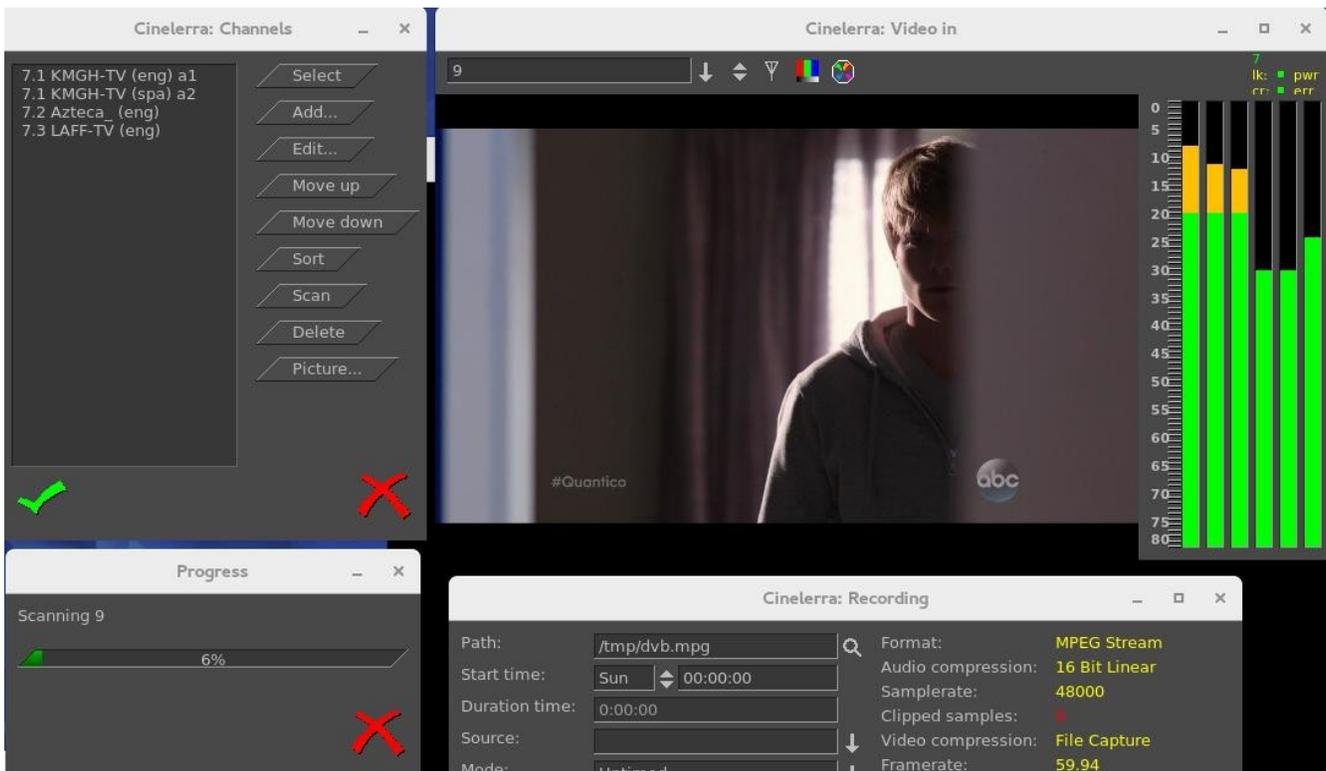
Scan the channels once the devices are setup by performing the following.

- 2) From the main window pulldown use: File → Record (lower case r) & resize windows. Two windows appear - the “Recording” and the “Video In”. The “Recording” window should have “Monitor video” checked and “Monitor audio” unchecked. Remember to check the Monitor audio later when you want to listen to a TV progra. On the “Video In” toolbar, click the antenna button (Edit channels tool tip). When the Channels window opens, click scan - “Scan confirm” appears. Make sure Freq table NTSC_DVB, Norm NTSC, and input0 are chosen automatically. Click OK and then wait for scan to end which may take several minutes.

Screenshot below on the left shows the Channels window and the popup waiting for scan confirmation. On the right side is the usual Recording window with the Video in window hidden behind.



Screenshot below shows scanning in progress. Note the “Video in” with the antenna (3rd symbol).



Any time after the channels have been scanned, you can edit them from the “Recording” window channel selection dialog (shown on left of screenshot above, such as “Sort”). You do not have to scan the channels every time as they are saved in a database referred to as the channel-db.

Click chk-OK when done scanning channels and making any edits you want.

After the channel-db is ready, you can watch any of the selectable channels.

9.2.3 Watching TV using Cinelerra

If you have setup for DVB capture as outlined in the section above, just start here to watch TV.

From the main window use File → Record to request the “Recording” application.

In the “Video in” window, press the down arrow on the left to see the channel pulldown.

Select the desirable channel, and watch TV. You will have to enable the audio by checking the “Monitor audio” in the Recording window for sound.

Helpful hints: for any computer that you might choose to use that is a little slower, you can

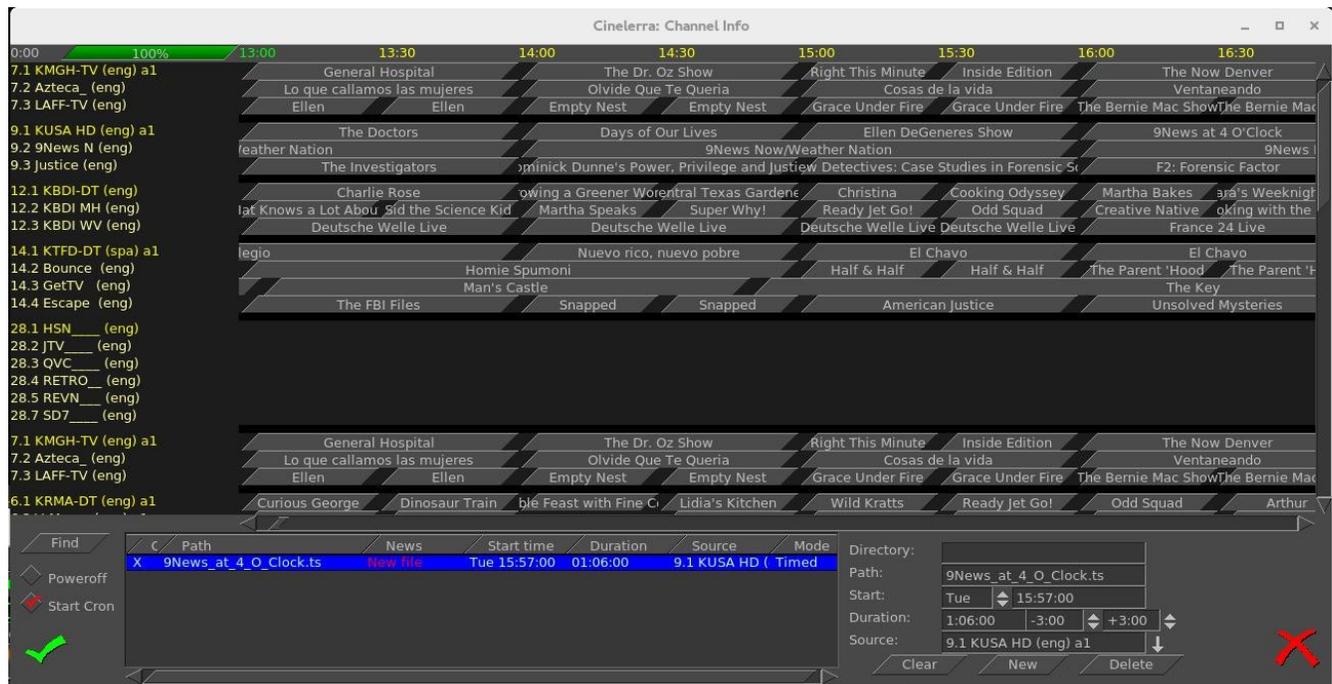
improve the available resources for watching TV by unchecking “Audio meters”, checking “drop overrun frames”, and checking “fill underrun frames”. Also, uncheck “Realtime TOC” in the Preferences → Recording window.

Dismiss the “Recording” window with chk-OK, or X to cancel when finished.

9.2.4 Recording TV to Capture at some Future Time via Batch

After the initial setup and channel scan completion, with the Recording application down (so that the /dev/dvb/adapters0 hardware is not in use) you can scan the DVB channel program data by pressing “Shift S” from the main window. It can take a minute or two to scan the channels. The resulting display is a large array of program buttons, with tooltips which are the TV program info text.

Screenshot showing Channel Info window after the “Shift-S” completes scanning TV program data.



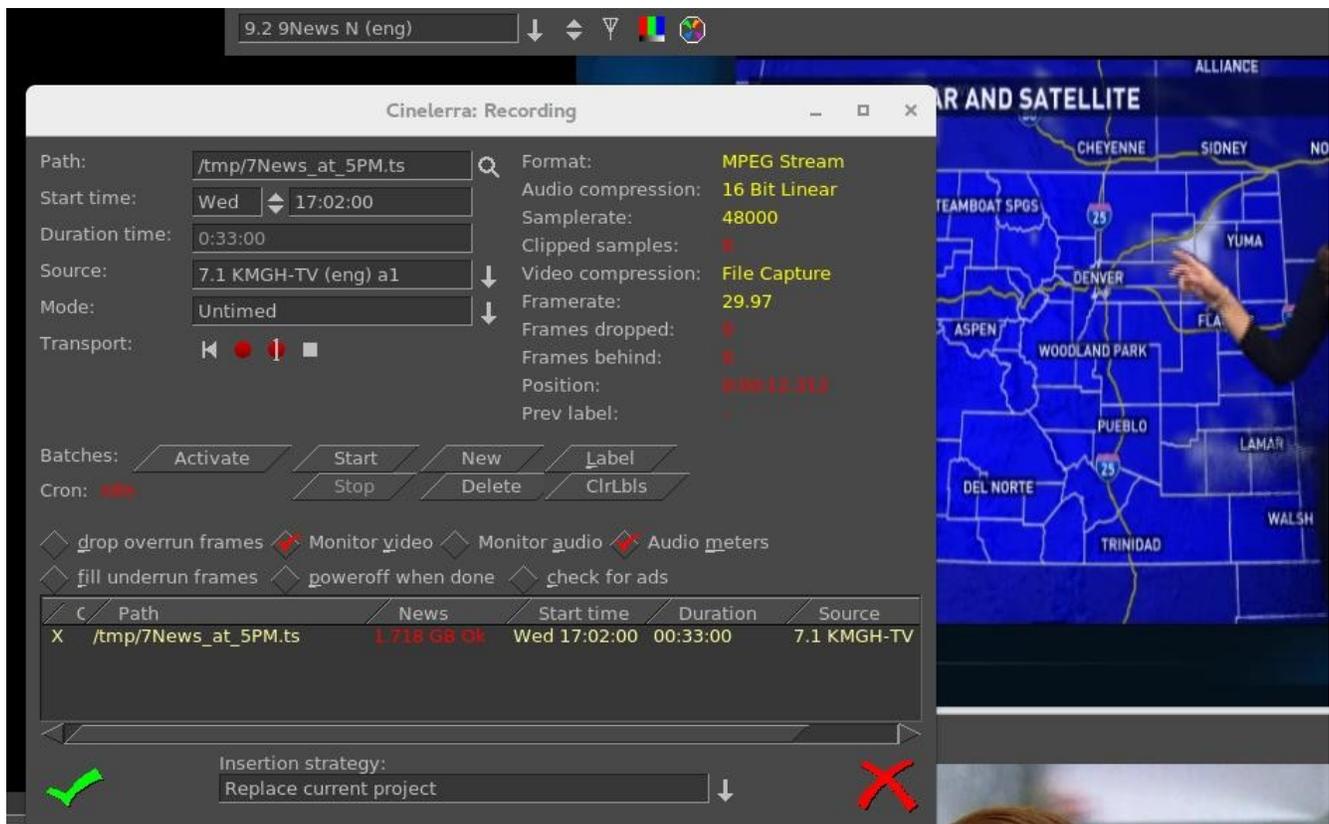
Now if you mouse over a program button, you see the TV program's info. Each of the programs is represented by a button. If you press a button, it preloads the program recording information into the batch record box below. In the bottom right corner box, you should set the “Directory” path of where to write the recording before pressing a button, so that the files will be written to an area of sufficient size to handle the recording (9.5Gb/hr). The two little spinboxes on the Duration line in the right bottom corner are leadin/leadout times to start recording a few minutes early, and end a few minutes late to prevent losing program endpoints. Instead of setting it to the original default of 0, you might want to start 3 minutes (-3.00) before and stop 3 minutes after (+3.00). Also, be sure your computer has the correct time or it will miss. Check to make sure the “Start Cron” option is checked (default). The batch job will start monitoring the channel to be ready for capture once you chk-OK and you will see the “Recording” window and the “Video in” window come up. The computer will power off after completion if the “Poweroff” checkbox is selected (not default). The job line of the active capture in the batch bay will flash continuously once the recording is started.

9.2.5 Recording Immediately to Capture a current TV program

If you have setup for DVB capture as outlined in the first section above, just start here to record a TV program that is currently playing and stop it whenever you want. Media files can be directly captured from the “Recording” window to include Start, Stop, and then loaded via chk-OK.

- Bring up the “Recording” window via the main window File → Record (lower case r).
- Set the “Path” to the write file Path; be sure there is enough disk space (9.5Gb/hr).
- Set “Mode” pulldown to “Untimed” found in the upper left corner area of the window.
- Press transport "start recording" button (red round button).
- Path will flash and update in batch window and remain flashing while recording.
- Press transport "stop recording" button (white square button).
- Before clicking chk-OK, be sure to set the Insertion strategy.
- Pressing chk-OK attempts to load any newly recorded media data for review..
- Or press X-cancel to terminate and dismiss the “Recording” application.

Screenshot showing the recording of a currently running broadcast TV program weather report.



9.2.6 Watching a Previously Recorded TV program

If you have setup for DVB capture as outlined in the section above, just start here to watch a previously recorded TV program. Use the following steps.

In the main window choose File → Load files and select the desired media file to watch.

Broadcast transport streams may consist of multiple programs that all run together on one channel (for example 9-1, 9-2, and 9-3 are all on channel 9). Therefore when you play the recorded program back all of these programs will play simultaneously and that is a jumble.

In order to select just the one you want, press Alt-1 in the main window to select Asset Program 1 (or Alt-1...8 to select the desired asset program number) which removes all of the unrelated audio/video tracks except for the selected program. It also resets the session to match the media.

Move timeline to where you would like to start watching.

Be sure the Compositor window is up and you will see the pre-recorded program there.

Use either the keyboard, a remote control that has been set up, or Android device for controls.

To toggle between fullscreen or original size, press “f” in the Compositor window.

To watch on a large TV screen or monitor which you have connected, there are numerous Remote Control keys to manipulate the playback as outlined in the next section.

9.2.7 Remote Control for DVB

Cinelerra DVB recording/playback can now easily be done in “couch potato” mode using a remote control. This comes in handy when you want to playback on a big TV screen for multiple person viewing. You can use the Application/Menu key on the keyboard to toggle between standard Cinelerra usage or a Dispatcher methodology through use of a remote control, an Android programmed device such as a tablet, or keyboard. The Application/Menu key on most keyboards is between the Alt and Ctrl keys on the right hand side bottom and depicts a menu on it either with or without a pointer (see image below). An ati-x10 Remote Control device is currently working with Cinelerra and other remote controls may work but have never been tried. The orange box on the top left main screen of Cinelerra indicates Dispatcher Method access is activated which allows for keyboard grab which routes all key strokes until toggled off. Below are the currently defined operations.

Remote Control Keys (Application/Menu key toggle for ati-x10 remote)

- up arrow - forward 1 minute
- down arrow - backward 1 minute
- left arrow - back 10 seconds
- right arrow - forward 10 seconds
- d - toggle channel scan menu
- f - toggle full screen
- a - select playback a and tile windows left (all on laptop)
- b - select playback b and tile windows right (all on TV)
- c - select playback c and tile windows left, but composer right (TV mode)
- 0-9 - select timeline by 1/10s from beginning to end
- a - toggle audio mute
- m - toggle audio meters
- keypad / - display mpeg ts stream program data
- keypad * - display mpeg ts stream channel schedule date
- keypad + - change channel, next
- keypad - - change channel, prev
- keypad . - start channel change by numeric entry
- keypad .. - clear channel entry
- book key - toggles channel scan (always available)
- hand key - terminates cinelerra (always available)

Plus usual Transport keys

- e - ??
- f - toggle full screen
- << - fast reverse
- > - toggles normal playback
- >> - fast forward
- red o - slow reverse
- square - stop
- 2 lines - fast reverse

Ati-x10 Remote



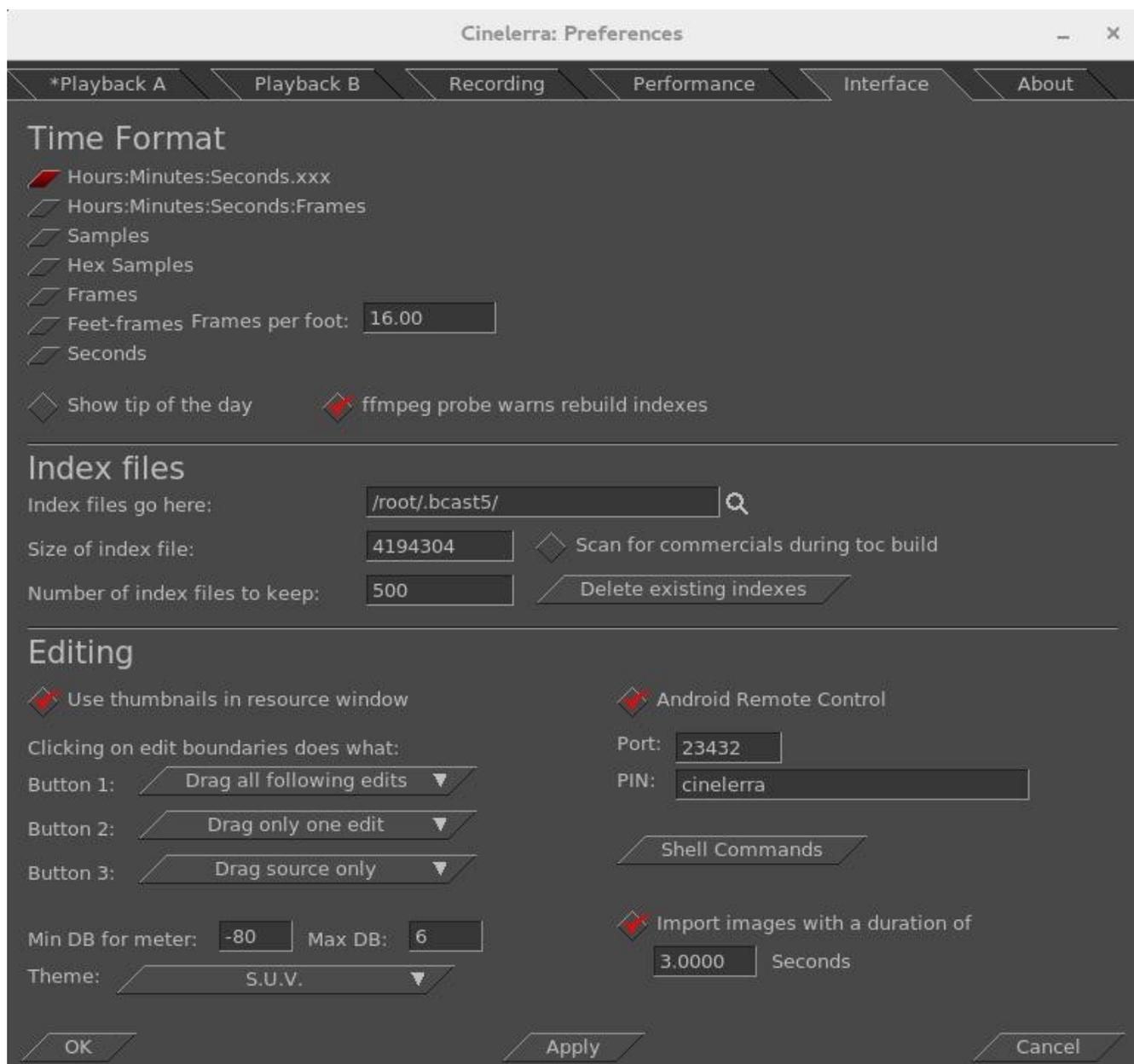
Application/Menu key
right bottom of keyboard



9.2.8 Android Remote Control for DVB

Cinelerra 5.x has android remote interface code. Any device, such as a tablet or a phone, can be used as long as it is running the Android operating system. Programming an Android Remote Control for DVB is a bit complicated at first, but becomes pretty simple after a bit.

- 1) The Android Remote Control requires you to download and install the android CineRmt app package (apk). To install it simply download it (any way you can) to your phone or tablet and click it to install, if it doesn't run automatically.
- 2) You should be able to open Settings → Preferences → interface tab. And see Android Remote Control in the editing section. Enable it using the checkbox. The default port 23432 and default PIN cinelerra are OK.
- 3) After you install the new phone/tablet app in step 1, start it up from your apps menu.



Before you can use it, the Android device must have the ip address of your computer entered into the configuration menu. The PC you are running Cinelerra on and the Android device have to be on the same network. You can tell it is in communication by tapping the Power (menu) button. When the remote is operated with the Power (menu) button, there is a little orange box on the upper left corner of the display to indicate the remote state is active. If you do not see the little orange box when the menu key is tapped, the communication is broken and must be fixed before proceeding. Follow the directions in the paragraph below.

BIG NOTICE: the firewalls in your computer and wifi router can stop this thing dead. Re-configuring a firewall is tricky and varies from distro to distro. If possible, during setup it is suggested that you disable the firewall temporarily. For those with advanced skills, use tcpdump or wireshark to look for udp messages from the IP address of the device.

SECTION NOT COMPLETE – (manualandroid)



Screenshot of an Android Tablet that can be used as a Remote Control for Cinelerra Recording and Playback.

9.2.9 How to Setup your Remote Control Device to work with Cinelerra

SECTION NOT DONE !! This section explains how you can take your own brand of remote control and set it up to easily work the way you want with Cinelerra.

9.3 The commercial DB... What is that? A TDB – Traveling Data Base

While watching broadcast television, it is possible to use Cinelerra to automatically mute commercials being aired during a broadcast capture. Visual muting as well as sound muting is done! You can NOT use ffmpeg for the format when working with the commercial DB. This database that was added is a “novel architecture” shared memory database, a Traveling Data Base (TDB). With TDB you have direct access to the shared database presence (instead of having to go to the server).

Basically it works as follows, but there are a few differences in just how it goes about it:

- 1) Identify a zone in a timeline as a commercial.
- 2) Characterize it concisely, in the audio and video timeline.
- 3) Store the characterization in a database.

Then, when capturing media or building a table of contents for an mpeg transport stream (broadcast capture DVB):

- 1) Characterize each I-Frame (a few frames per sec).
- 2) Search the database in real-time to identify the image.
- 3) If a match is identified, automatically press the mute button.
- 4) Unmute if the pattern match fails, or the commercial ends.

It mostly works, but it takes more time to setup and maintain the database as the advertisements change frequently or vary by a small amount. It would take a small set of personnel to keep the database current and some finessing of the code.

Currently, to activate the commercial database (db) methods you must do the following as root:

```
cd <cin-path>/bin
mkdir /cinelerra          (yes, currently only in the / filesystem)
./new_db /cinelerra/media.db (to create a new empty commercial capture database)
export BC_USE_COMMERCIALS=1 (in the command shell that is starting cinelerra)
```

This activates the gui tools and database access. As of yet, all of the requisite permission changes necessary to run as non-root have not been worked out (initially just easier to run as root).

When a commercial is "cut", the following occurs:

- 1) transmits the capture timeline fragment to /tmp
- 2) scans the snip to characterize it statistically
- 3) creates a miniature b/w movie of the beginning and end
- 4) stores all of this into the media.db

There are 2 ways currently to identify a commercial:

- 1) Load mpeg3 transport media, highlight select the commercial in the timeline, and press "cut commercial" on the toolbar.

- 2) Use the remote control interface to indicate the boundary in/out cut points in the realtime capture timeline, and operate a cut command.

The cut command creates a popup to let you know that it is processing the captured data. This can take several seconds depending on the size and media format. The results from the scan are stored in a new record in the “media.db”.

The stats are somewhat disappointing. Image recognition is way past zero, it is not 100 percent. Broadcast streams have tons of minor variances which hamper digitally perfect reproduction from instance to instance of commercial presentation. This puzzle remains to be solved.

To review media stored in the “media.db”, and delete old expired commercial data records, press (Shift-M) from the main window. This will create a media-db window which shows the various captured samples stored there.

Currently, only a customized ati-x10 (kernel mod required) radio remote control works. But this can quite readily be modified for other remote control models.

10. Themes User Interface

Akirad themes have all been added to Cinelerra 5.1 with major thanks to the Cinecutie project for all of their hard work. These modifications provide alternative User Interface environments and allows you to choose your favorite look. Be aware that when you change your theme, Cinelerra will automatically shutdown and restart so that it takes effect. The theme you choose is preserved across sessions.

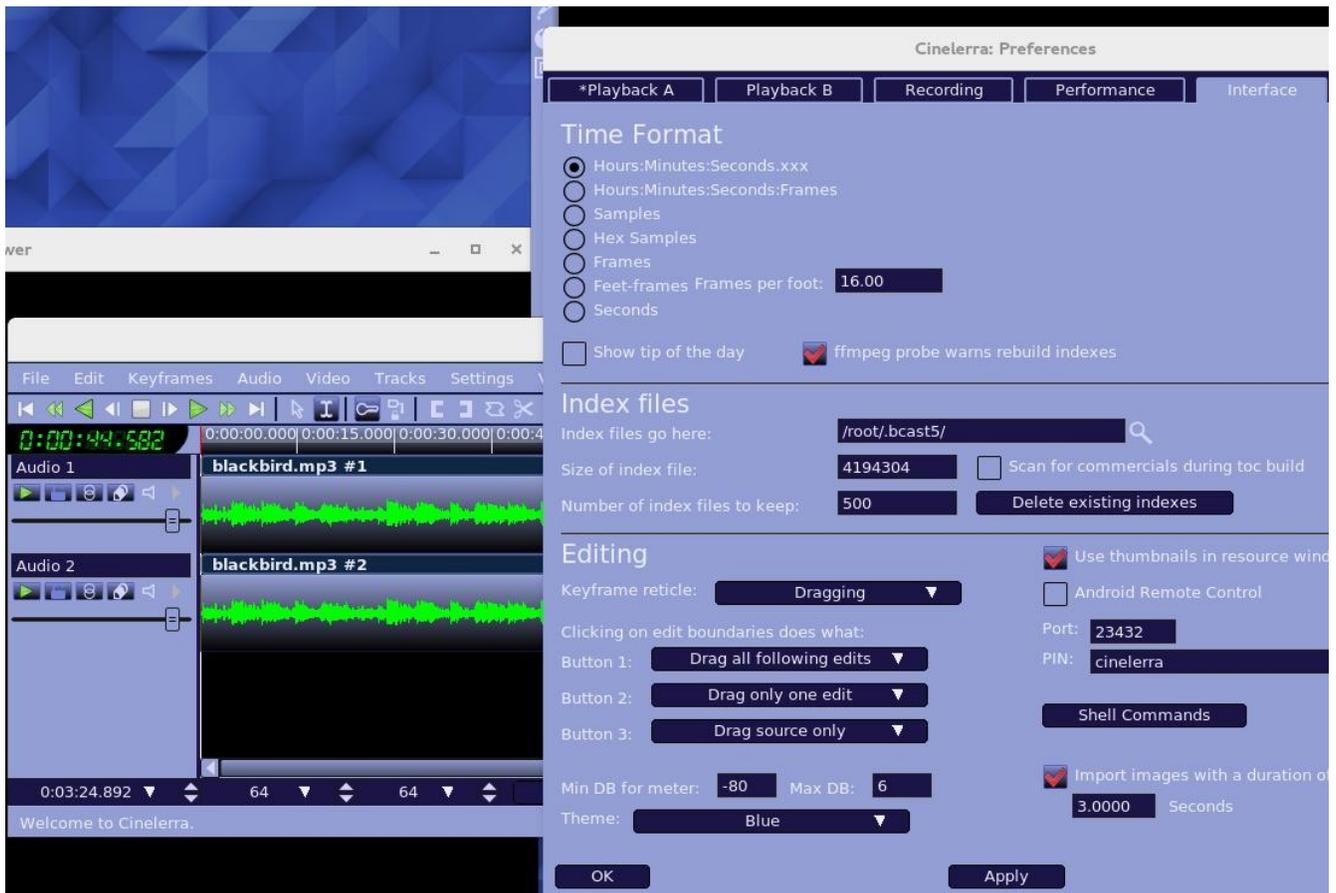
Theme choices available:

Blond
Blond-cv
Blue
Blue Dot
Bright
Hulk
Pink Lady
S.U.V. (default)
Unflat

How To Change your Theme:

To change a “Theme” in the main window pulldown, select Settings → Preferences then in Preferences window, click on the “Interface” tab. In the Editing section in the lower left hand corner, click on the down arrow next to Theme to see your choices. Click on your desired choice from the list given. Check OK, cinelerra will automatically shutdown and restart.

Screenshot showing the “Blue” Theme and the Preferences Window where you can change Theme.



11. Resizable Fonts/Icons Support

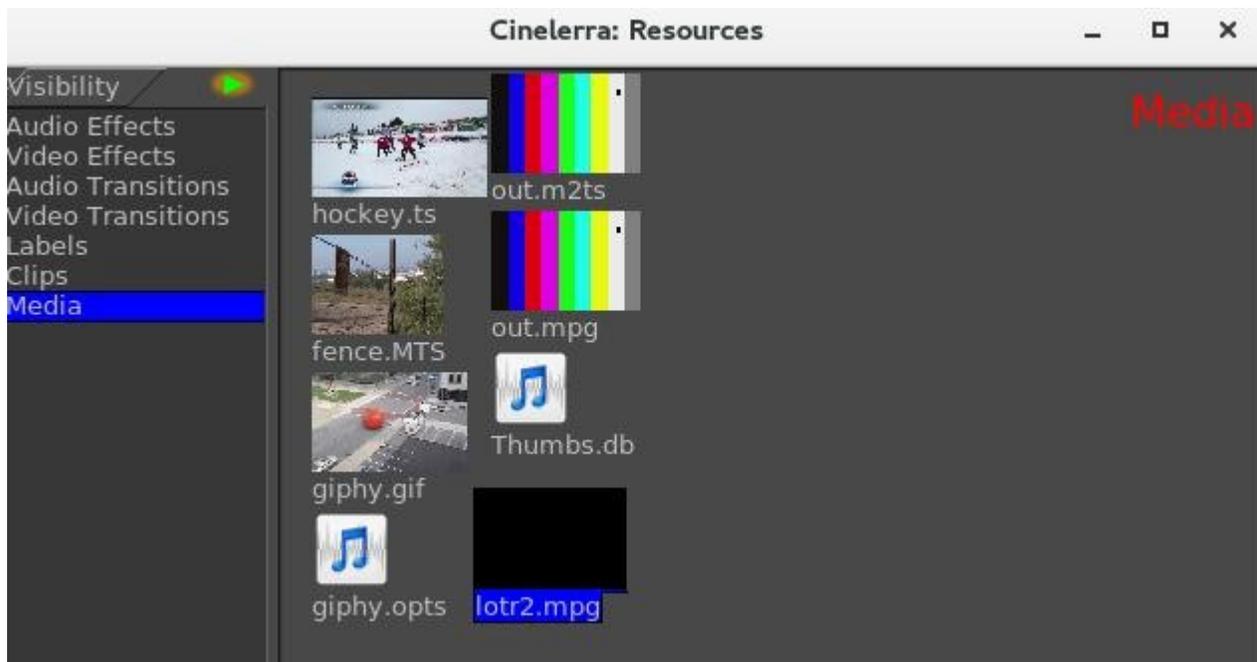
You can increase the size of the characters in the fonts and icons on your Cinelerra system. This will make it easier to read the characters if you have trouble seeing the default small letters, which have been auto-scaled based on the window geometry. The user-friendly font/icon scaling default is 1. To defeat default auto scaling and get any size characters/fonts, override the setting via shell environment variables. For example:

```
export BC_FONT_SCALE=2.0 ! twice as big, may show letters beyond the borders
export BC_ICON_SCALE=1.5 ! 1.5x bigger
```

12. Vicons – aka Video Icons

Vicons are video icons. The Resources window now will play the first 5 seconds of video looped in the area occupied by the video media icons. This is only enabled for the "media" folder in "icon" mode when the mouse pointer is actually inside the Resources window. There is also a "playback" toggle to enable/disable the icon playback activity. This playback toggle is a green arrow to the right of the Visibility label as seen in the screenshot below. It is active when surrounded by a gold background and inactive if no background. When it is enabled/active, if you click on one of the vicons, a view popups that increases the size to 4 times larger. This makes it easier to see if it is the video you are looking for in case you have many similar media files. To conserve memory, the video is stored 8-bits per pixel which results in low image quality.

Screenshot showing the Vicons in the Resources window with the Playback toggle “active”.



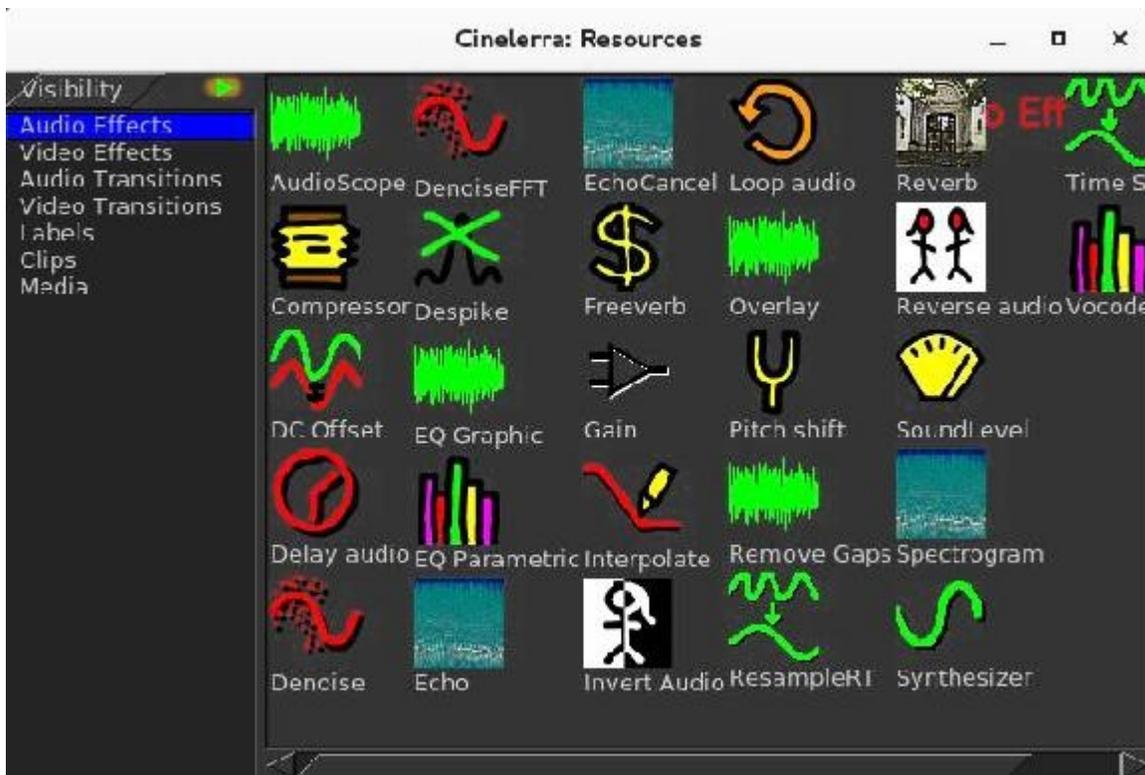
13. Toolset Visibility Support

Cinelerra contains many plugins, especially with the recent addition of ffmpeg, and it is somewhat difficult to find the one you are looking for in the Resources window. In 5.1, the plugins have been categorized into the following subsets in the Visibility section of the Resources window to make it easier to locate a particular one:

Audio Effects, Video Effects, Audio Transitions, Video Transitions, Labels, Clips, Media

Assistance in further categorizing these plugins would be welcome. There may be additional categories needed or plugins that are currently miscategorized for a more intelligent grouping.

Screenshot showing on the left hand side the Visibility box with Audio Effects highlighted.



Another helpful feature of the “Visibility” toolset in the Resources window gives you the ability to turn off or on any of several sets of plugins. If you left-click the Visibility box, you will see the various categories of plugins, such as ladspa, ffmpeg, colors, transforms, etc. Highlight the set you want to turn on and a check mark appears to show it is active. Highlight again to toggle it off. And handiest of all is the ability to add your own personal directory of plugins which will show up here. See the attached screenshot which illustrates that all of the plugins are turned off (not visible) except for your own personal set, called “old_plugins”, which just contains the AgingTV plugin. All you have to do to have these plugins become visible is to create a directory, with some name that is meaningful to you, and put your .png files in your cinelerra_path bin/plugins/<your_directory_name>.

Screenshot showing the Visibility categories of plugins with all toggled off except old_plugins.



14. Menu Bar Shell Commands

In order to provide some configuration control, the Menu Bar Shell Commands are available for customization purposes. In the main window on the top line containing the File, Edit, ... Window pulldown menus, all the way to the right hand side is the “shell cmds” icon. You will see a small gold-color bordered box with the >_._ inside and if you mouse over it, the tooltip says “shell cmds”. This is a configurable popup which gives you the ability to execute a pre-defined script to do something such as post processing or whatever needs to be done on a repeating basis.

The Shell Cnds popup menu items are configured in:

Settings->Preferences->Interface->Shell Commands

This runs a dialog that lets you Add/Del/Edit the items in the main menu popup. Possible options are:

1) Add a new menu item/script:

click Add

in the Label box, change the word "new" to the desired menu label

in the Commands box, type the command lines to be included in the script

(for example: Label => ls; Commands => ls -l

check-OK in the Commands window

check-OK in the Shell window

click on OK or Apply in the Preferences window

2) Del to delete an existing menu item (for example “manual”):

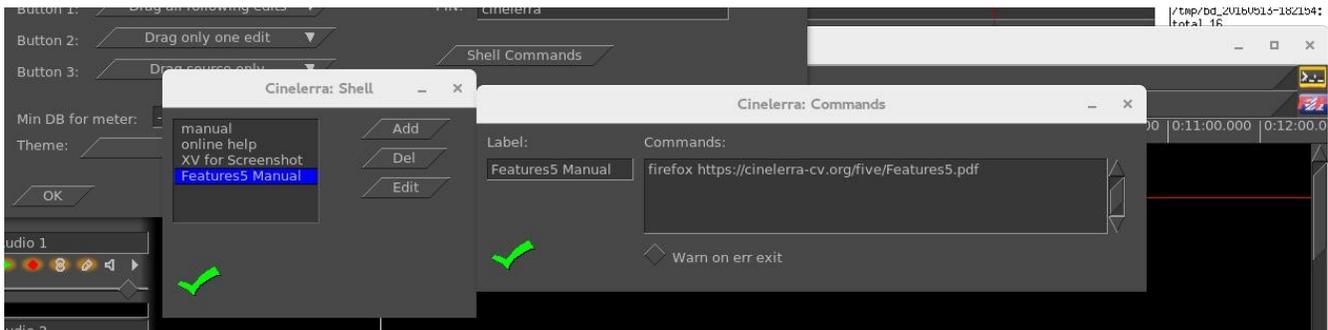
select to highlight an entry in the Cinelerra: Shell listbox

click Del

check-OK in the Shell window
click on OK or Apply in the Preferences window (the entry is destroyed)

- 3) Edit an existing menu item:
click Edit
select to highlight an entry in the Cinelerra: Shell listbox
click edit (the edit dialog appears)
set the label and/or commands to the new desired values
check-OK in the Shell window
click on OK or Apply in the Preferences window

Next time you click on the “shell cmds” icon, you will see your new labels.



15. Last Play Position Memory

When you play media, the start/end playback positions are saved as if they had been made into temporary labels. They appear on the timeline as purple/yellow hairline markers representing the last start/end labels for the last playback. They can be addressed as if they are label markers using:

- Ctrl ← tab to the label before the cursor, that is “play start”
- Ctrl → tab to the label after the cursor, that is “play stop”

You can use these markers for re-selection. Additionally, the selection region can be expanded by “pushing” the markers using single frame playback. Use frame reverse (keypad 4) to push the start play marker backward, or use frame forward (keypad 1) to push the end play marker forward.

Another handy feature is to use the combination of Ctrl-shift-arrow (left or right) to select the media from the cursor position (red hairline) to the start or end marker by “tabbing” to the label markers. For example, tab to the beginning of the previous play region using Ctrl-left-arrow to move the cursor to the beginning of last play, then press Ctrl-Shift-right-arrow to tab to the end of the playback region. Now you can clip/play/expand or edit the previous playback selection.

- Ctrl SHIFT → tab cursor to label right of cursor position and expand selection
- Ctrl SHIFT ← tab cursor to label left of cursor position and expand selection

16. Audio H/W Gain Support

This feature, device level gain, corrects for hardware conditions which some devices may need to be useful. For example, you may need to increase the gain for a weak microphone or a noisy speaker, since it affects rendering when you crank up or down the audio via use of the patchbay. With the audio H/W gain support, you have the ability to fine tune the audio volume by some numerical value for the scale. You are adjusting the scaling of data into the audio driver - H/W scaling is done before it goes into or out of the driver. This is a one time linear multiplication of the sample values, and may offer better control than the logarithmic DB gain controls of the application. These device controls are in Preferences:

Settings → Preferences → Playback(A/B) → Audio Out → Gain

Settings → Preferences → Record → Audio In → Gain

17. Valgrind Support Level

Valgrind is a memory mis-management detector. It shows you memory leaks, deallocation errors, mismanaged threads, rogue reads/writes, etc. Cinelerra 5.1 memory management is designed to work with Valgrind detection methods. This assists in developing reliable code. Use of Valgrind points out problems so that they can be fixed. For example, when this version of Cinelerra shuts down, it deallocates memory instead of just stopping, thus making memory leak detection possible.

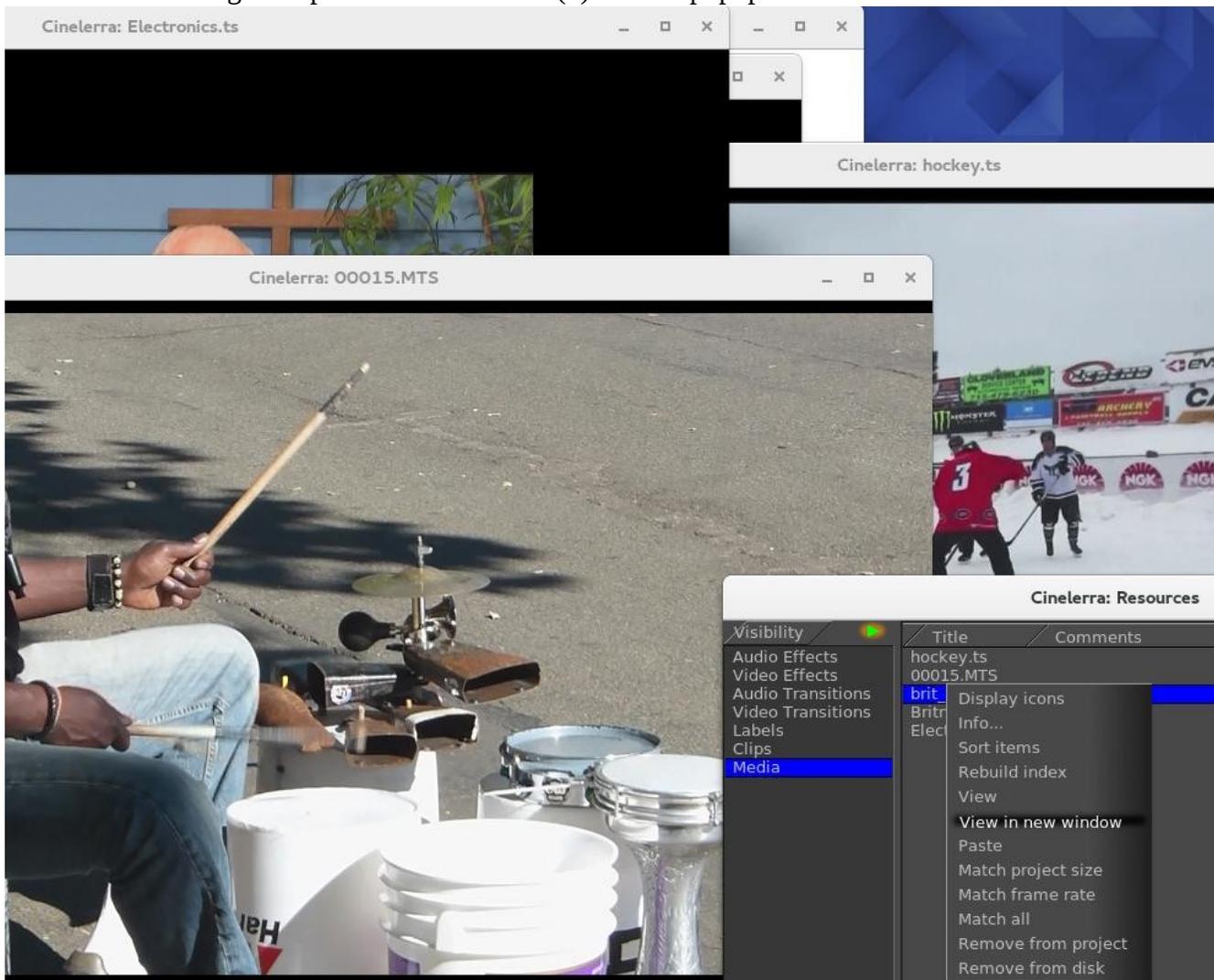
18. CFLAGS has -Wall

When compiling Cinelerra 5.1, a CFLAGS option used is “Wall” where the “W” represents warnings and “all” means all. This causes the compile to check for simple mistakes that can be detected automatically and issue warnings when the code is questionable. It can also detect situations where the compiler will generate incorrect code, like type-punned pointer. By turning on this flag, when new code is vetted for predictable mistakes, the code can be corrected before becoming manifested in the application.

19. Multi-Viewer Window Support

You can create as many Viewer windows as you want in 5.1. These are handy for users who are adept at working with a lot of different clips simultaneously. By bringing up multiple Viewer windows, each clip can be edited in its own area, making it easy to see all of the separate pieces. After you have loaded some media files, to start another Viewer window, right click on one of the pieces of media in the Resources window. This brings up a menu of several options, one of which is “view in new window”. Choose this option and that media will come up in a new Viewer window for you to work.

Screenshot showing multiple Viewer windows (3) and the popup menu with “View in new window”.

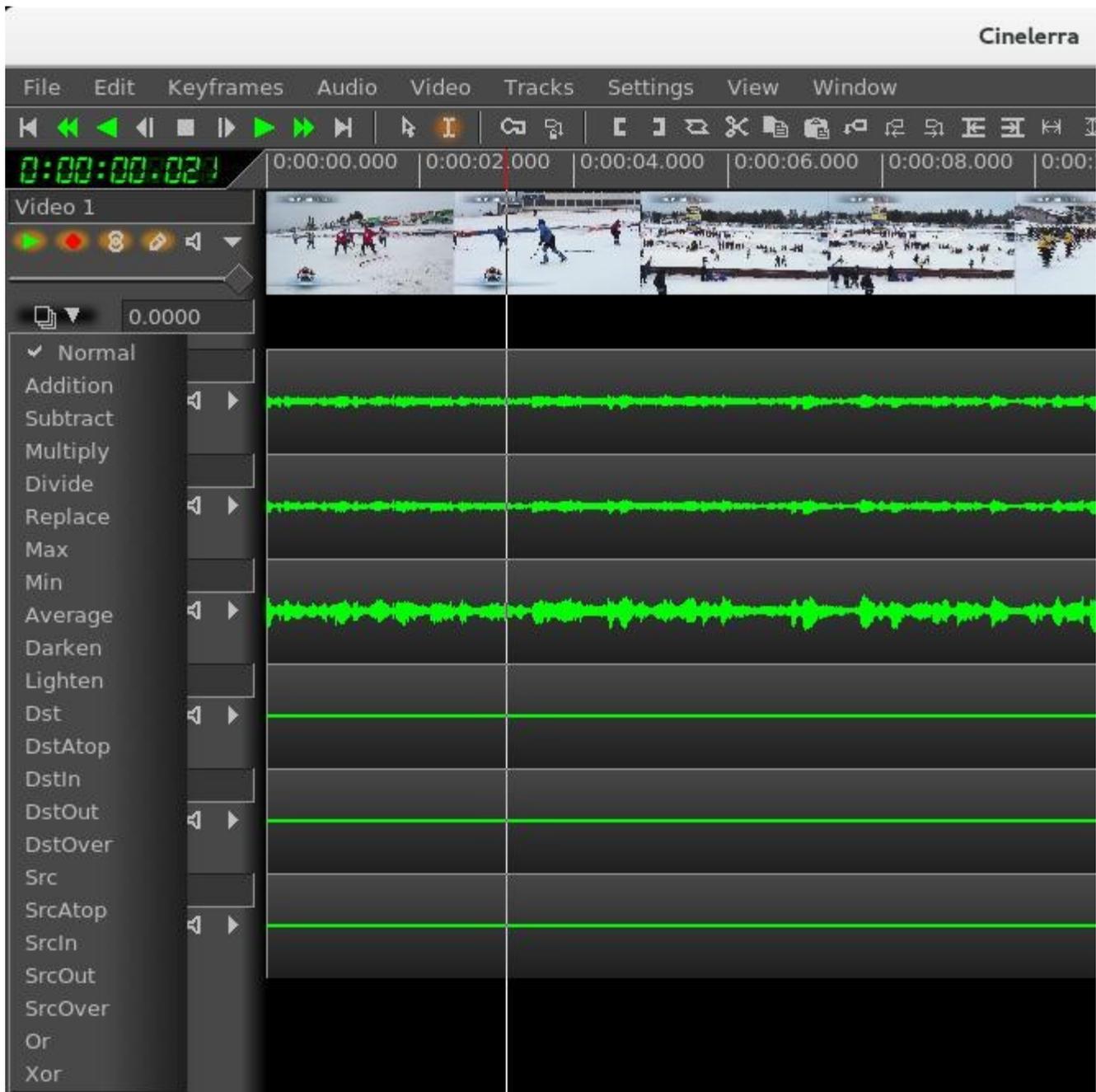


20. Porter-Duff Operators and Overlay Modes

The purpose of the Porter-Duff operators is to control the foreground and background stacking and use blending to reshape image object boundaries. It normally makes use of a binary type alpha blending system for all in or all out. To use the available operations in Cinelerra, follow these steps:

- 1) In the main window, look at the Patchbay on the far left.
- 2) Click on the small arrow that points to the right, inside the panel to see it expand.
- 3) A down arrow shows up with a tooltip of “overlay mode”.
- 4) Use the down arrow to get to the popup menu and choose the desired effect from the 23 possibilities. “Normal” is the default.

Screenshot showing the 23 Porter-Duff operators/overlay modes in the Pulldown on the left side.



Porter-Duff is the industry standard for alpha blending operations. Only a short explanation follows here, but there is much more information to be found on the internet with complete descriptions and examples. Alpha blending is the process of combining a foreground color with a background color which produces a new blended color. The “alpha” channel describes how much opacity is present in a pixel. It may be completely transparent, completely opaque, or any range of translucency.

Conceptually, when the foreground color is completely opaque, the resulting blended color will be the foreground color. If it is transparent, the blended color will be the color of the background. Any range of translucency in the foreground/background colors is handled formulaically.

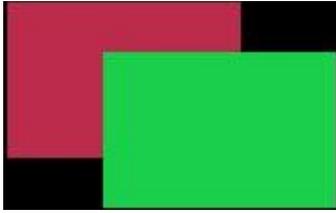
When the value of the alpha channel is 1, the image is all there, if it is 0, there is no image at all, otherwise it is only partially there. In other words, the alpha value goes from 0 to 1, where full transparency is 0 and opaque is represented by 1. Alpha blending models opacity.

When blending source and destination shapes, the shape boundaries can be changed with the alpha blending effects. There are 4 regions: only the Source is present, only the Destination is present, Both are present or Neither is present. Various “blending” effects are created by determining what to do in each of the 4 areas. There are a total of 12 standard Porter-Duff operators, but there are 23 possible overlay modes used in Cinelerra 5.1. Each is characterized by its value in the three regions: source, destination and both, with the “neither” region always being blank. The source and destination regions can either be blank or filled with the source or destination colors. A specific compositing algebraic formula is used to calculate effect. The implementation math forms are listed here, where D=Destination, S=Source, a=alpha, c=chroma (color). Each line describes a pair with the left one for alpha and the right one for chroma.

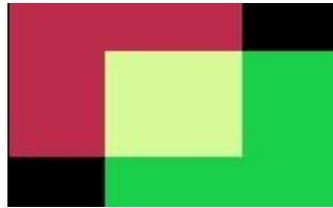
| | |
|----------|--|
| DST | [Da, Dc] |
| DST_ATOP | [Sa, Sc * (1 - Da) + Dc * Sa] |
| DST_IN | [Da * Sa, Dc * Sa] |
| DST_OUT | [Da * (1 - Sa), Dc * (1 - Sa)] |
| DST_OVER | [Sa * (1 - Da) + Da, Sc * (1 - Da) + Dc] |
| SRC | [Sa, Sc] |
| SRC_ATOP | [Da, Sc * Da + Dc * (1 - Sa)] |
| SRC_IN | [Sa * Da, Sc * Da] |
| SRC_OUT | [Sa * (1 - Da), Sc * (1 - Da)] |
| SRC_OVER | [Sa + Da * (1 - Sa), Sc + (1 - Sa) * Dc] |
| OR | [Sa + Da - Sa * Da, Sc + Dc - Sc * Dc] |
| XOR | [Sa * (1 - Da) + Da * (1 - Sa), Sc * (1 - Da) + Dc * (1 - Sa)] |
| NORMAL | [Sa + Da * (1 - Sa), Sc * Sa + Dc * (1 - Sa)] |
| ADDITION | [(Sa + Da), (Sc + Dc)] |
| SUBTRACT | [(Sa - Da), (Sc - Dc)] |
| MULTIPLY | [(Sa * Da), (Sc * Dc)] |
| DIVIDE | [(Sa / Da), (Sc / Dc)] |
| REPLACE | [Sa, Sc] (fade = 1) |
| MAX | [max(Sa, Da), MAX(Sc, Dc)] |
| MIN | [min(Sa, Da), MIN(Sc, Dc)] |
| AVERAGE | [(Sa + Da) * 0.5, (Sc + Dc) * 0.5] |
| DARKEN | [Sa + Da - Sa*Da, Sc*(1 - Da) + Dc*(1 - Sa) + min(Sc, Dc)] |
| LIGHTEN | [Sa + Da - Sa*Da, Sc*(1 - Da) + Dc*(1 - Sa) + max(Sc, Dc)] |

Below are the results of utilizing the 23 available operations within Cinelerra as listed on the previous page. Src is the solid green rectangle and Dst is the solid red rectangle. There are better illustrations of what alpha blending can do, however for consistency sake, these are the results when using standards.

normal



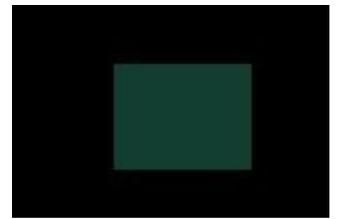
addition



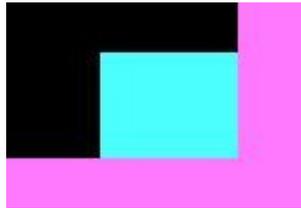
ubtract



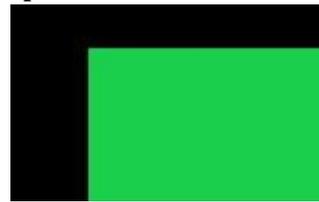
multiply



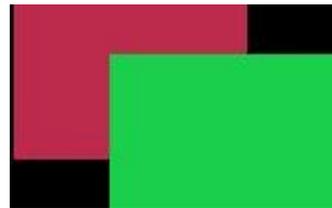
divide



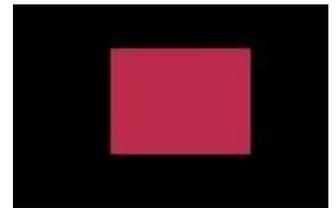
replace



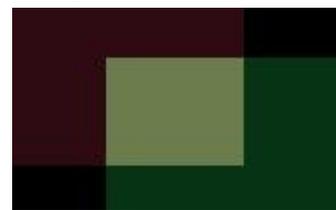
max



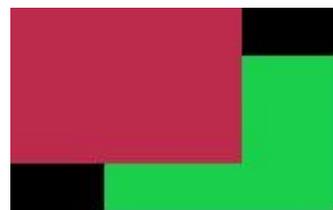
min



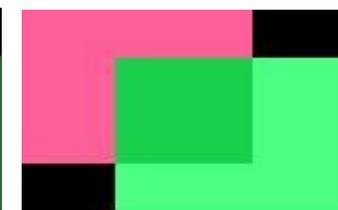
average



darken



lighten



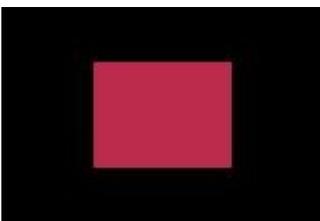
dst



dsatop



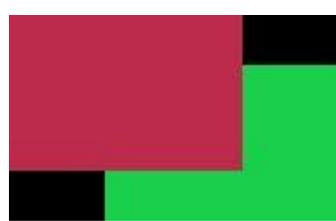
dstin



dstout



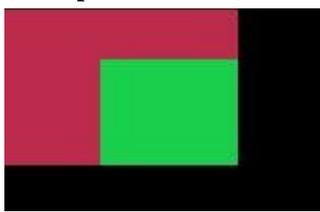
dstover



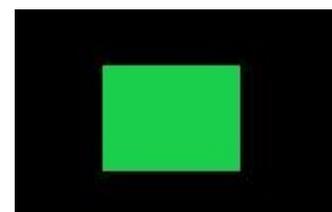
src



srcatop



srcin



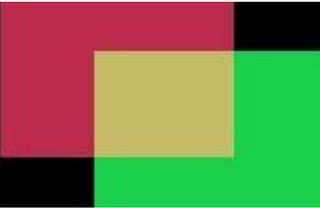
srcout



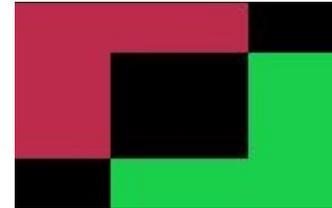
srcover



or



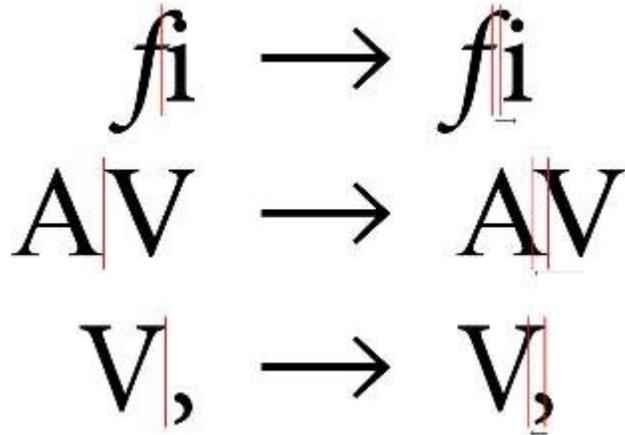
xor



21. Titler Font Kerning

When using the Titler, kerning is applied in order to allow parts of a letter to go outside the standard sized letter box. Kerning is the process of adjusting the space between individual letters. It is not the same as proportional spacing. In kerning, a bounding box is allowed to overlay another bounding box. The philosophy here is to aim at the ability to have boxes overlap in order to make the letters look more visually appealing. Bounding box and escapement are tracked separately. In addition to adjusting individual letter spacing, the program will also expand the render box in order to have any parts of the letter extend outside the standard letter box. Kerning is applied to any and all fonts.

Screenshot shows how letters can be presented without and with kerning. Note this is just an example.



APPENDIXES

Appendix A: Crash Handling Tips

This section is a handy guide for describing various kinds of software computer system failures. Only some of these various lockups or crashes can be dealt with. Hopefully, it will help to have some hints to know what kind of failure it is, or to save your work or to avoid future problems. For most of this, your name must be root, although you can certainly try to see if it works for you when not root.

System lockups:

When the system locks up, it is usually a system problem. Normally an application program cannot hose the system. It is a major goal of system design to prevent an application (app) from failing a system interface. This does not mean an app can't mess things up.

Cinelerra crash:

This is covered in Section 6 – Crash Dumps for Analysis. Just a reminder that you must be root and by providing a crash dump and as much other information as possible, you will be helping the developer to analyze the problem and fix it so that it can be avoided in the future.

X Server crash:

Keyboard does not respond, screen is frozen, caps lock may operate LED light. Sometimes using ctrl-alt-F1 ... ctrl-alt-F7 (etc.) will allow you to regain control of a VT console. You can use this to login and check logs: eg. /var/log/Xorg.0.log, dmesg, journalctl ... etc. If you have another computer, make sure a terminal server is configured (for example: rsh, ssh, or telnet), then remote login via this other computer and check the logs. Most important is to immediately note the current software state, and the very last thing that preceded the crash, i.e. last button click, last keystroke, ... or whatever.

Kernel crash:

The machine goes completely dead. The keyboard caps lock LED will probably be flashing. Most likely the only way to see anything after the kernel crashes is to use a serial port console log and usually kdb, the kernel debugger, and special cabling. This requires a lot of setup, and is normally reserved for experts. Login from another computer will not work. Pinging the ip address will not respond since the network stack is part of the kernel. There are some virtual machine setups that will let you debug a “guest” kernel, but this also requires a lot of setup, and affects which kernel is currently under test. The kdb route is preferable.

Keyboard grabs, Server grabs, and Deadlocks:

A grab is an X-server state where all events are forced to just one window event stream. This forces the user to respond to the dialog. Things seems to be working, but no keypresses do anything useful. The system clock and other programs will still be working. The network will work for remote logins. Grabs can be canceled if the /etc/X11/xorg.conf X config contains special setup as shown below:

```
Section "ServerFlags"
    Option "HandleSpecialKeys" "Always"
    Option "AllowDeactivateGrabs" "True"
    Option "AllowClosedownGrabs" "True"
EndSection
Section "InputDevice"
```

```
Identifier "Keyboard"
Driver "evdev"
...
Option "XkbOptions" "terminate:ctrl_alt_bksp"
Option "XkbOptions" "grab:break_actions"
EndSection
```

or to \$HOME/.xinitrc, add:

```
# xkb terminate/grab actions disabled in xorg.conf, use:
setxkbmap -option "grab:break_actions"
setxkbmap -option "terminate:ctrl_alt_bksp"
ctrl-alt-bksp = terminate the X-server, may restart automatically
```

If your system is running in runlevel 5 (you might be at runlevel 3 for development), it also allows the user to cancel a keyboard grab using:

```
ctrl-alt-keypad-* = ungrab
ctrl-alt-keypad-/ = ungrab, kill grab owner process
```

Modal forms (always on top, and usually ptr/kbd grab) dialog boxes can hose a system by putting a form over another form holding a grab. This means the form that needs input may never get any, and the effect is a deadlock. Usually you will have to restart X (ctrl-alt-bksp).

Window Manager issues:

The "desktop" window manager can intercept and modify all kinds of user input. Mostly, this is a good thing, but can be a nuisance. If user keypresses can be programmed to trigger actions, then they may be useful to send KILL or INTR to an app that seems to be holding X's attention. For example:

```
killall -INTR cinelerra, killall -9 cinelerra,
killall X, but you must run as root to be able to do this
```

The ALT and META keys may be intercepted by the window manager, and this can cause unexpected interface operation.

