Internet Radio Automation and Encoding Toolkit

Kevin C. O’Kane
kc.okane@gmail.com
okane@cs.uni.edu
https://www.cs.uni.edu/~okane
http://threadsafebooks.com/
Table of Contents

1 Overview ................................................................................................................................. 7
  1.1 Status ................................................................................................................................. 7
  1.2 Required Hardware and Software ....................................................................................... 7
    1.2.1 Operating System Requirements .................................................................................. 7
    1.2.2 Hardware Requirements .............................................................................................. 7
    1.2.3 Window Manager .......................................................................................................... 7
    1.2.4 Other Running Windows .............................................................................................. 8
    1.2.5 Graphics Support .......................................................................................................... 8
    1.2.6 Hardware Timing .......................................................................................................... 8
  1.3 Goals ................................................................................................................................... 9
  1.4 Basic Design ....................................................................................................................... 10
  1.5 Performance ....................................................................................................................... 10
  1.6 Workspace Switching ......................................................................................................... 11
  1.7 Implementation .................................................................................................................. 12

2 The Automation Console ........................................................................................................ 12
  2.1 Overview ............................................................................................................................. 12
  2.2 Audio / Video Players ......................................................................................................... 15
  2.3 Browsers ............................................................................................................................. 15
  2.4 Console Layout ................................................................................................................... 15
  2.5 Console Placement ............................................................................................................. 17
  2.6 Casting Panel Buttons and Boxes ...................................................................................... 17
    2.6.1 File Select, Randomize, Text Entry and Play ................................................................. 17
    2.6.2 YouTube Videos ............................................................................................................ 19
    2.6.3 Full Directory Select and VLC ..................................................................................... 20
    2.6.4 Playlist Select, Shuffle and Play .................................................................................. 20
    2.6.5 File Search .................................................................................................................... 21
    2.6.6 Program Search .............................................................................................................. 22
    2.6.7 ffmpegV and ffmpegI .................................................................................................... 22
    2.6.8 OBS Button .................................................................................................................. 22
    2.6.9 Panel Button .................................................................................................................. 22
    2.6.10 Full Screen Button ...................................................................................................... 22
    2.6.11 Normal Resolution Mode ............................................................................................ 23
    2.6.12 Program Buttons ......................................................................................................... 23
    2.6.13 Seq ProgramButton .................................................................................................... 23
    2.6.14 Rand Program Button .................................................................................................. 23
    2.6.15 Stop Players Button ..................................................................................................... 23
    2.6.16 Pause Button ............................................................................................................... 24
    2.6.17 Origin Button ............................................................................................................... 24
    2.6.18 Next Button ................................................................................................................ 24
    2.6.19 Prog < Button .............................................................................................................. 24
    2.6.20 Prog > Button ............................................................................................................... 24
    2.6.21 Skype .......................................................................................................................... 24
    2.6.22 Saver Button ................................................................................................................. 24
Illustration Index

Figure 1 Automation Console .............................................................................................................13
Figure 2 Automation Console With VLC Video .......................................................................................14
Figure 3 Console in Panel Mode with Camera Feed .................................................................14
Figure 4 Alternative Casting Area Card................................................................................................16
Figure 5 File Chooser Example .............................................................................................................18
Figure 6 Full Panel YouTube Display ...................................................................................................20
Figure 7 File Search Example ...............................................................................................................21
Figure 8 Cam Inset Example ...............................................................................................................26
Figure 9 Example Search Box Results ...................................................................................................28
Figure 10 Screen Saver Mode .................................................................................................................30
Figure 11 Slide Show Mode .....................................................................................................................31
Figure 12 Skype View ............................................................................................................................32
Figure 13 YouTube View .........................................................................................................................33
Figure 14 OBS View.........................................................................................................................35
Figure 15 Camera View....................................................................................................................36
Figure 16 Automation.glade.............................................................................................................47
Figure 17 Linux Audio....................................................................................................................51
Figure 18 Sound Preferences..........................................................................................................53
1 Overview

1.1 Status
This is an on-going project. If you find errors or have suggestions, send them to the email address given on the title page. You may want to check back for updates from time to time.

1.2 Required Hardware and Software

1.2.1 Operating System Requirements
This system has been developed and tested on Linux Mint 18.3 with Mate. It also appears to work with Linux Mint 18.3 Cinnamon and Linux Mint 18.3 XFCE with appropriate adjustments to the themes and windows (Mint X theme and Metacity + Compositing).

Since the system makes use of window manager facilities, it may not work correctly on other versions of Linux with other window managers. It does not appear to work with Ubuntu 18.04 but may with other Linux distributions with a GTK 3+ based window manager. If not, the display will not be correctly organized and buttons will overlap on another.

1.2.2 Hardware Requirements
Due to the large number of threads and processes required to effectively run this system, a quad-core machine is highly desirable.

Overall hardware load can be minimized, if needed, by not running screen savers, slide shows, and cameras as well as using Normal Mode (reduces compressor requirements).

1.2.3 Window Manager
This software is configured to run with Linux Mint 18.3 with Mate, and, with appropriate adjustments, Cinnamon, and XFCE. Due to differences in GTK usage, the software may not work correctly on other distributions of Linux.

For proper operation, it depends on certain window manager features. If these are not present, all or part may not operate.

The following is a list of some Mint 18.3 Mate window manager options. They may be set in:

preferences | look & feel | desktop settings | windows

when using the Mint Custom Menu Bar.

The following options do NOT work:
If you use the above options, although you will see video on your screen, the video will not encode correctly when you use ffmpeg.

The following options DO work:

- Marco + Compositing
- Marco + Compton
- Metacity + Compositing
- Metacity + Compton
- Compiz

Please adjust your settings accordingly.

### 1.2.4 Other Running Windows

The system makes use of `wmctrl` to control window visibility. This program makes use of window names to control window behavior. You can see a list of currently running windows on your system by running the command:

```
wmctrl -l
```

When you run this system, you need to close all other windows first as the names of some of these may conflict with or duplicate that names of windows that `wmctrl` will control.

### 1.2.5 Graphics Support

Generally speaking, performance will be improved if your Linux system has access to a supported graphics card. Generally speaking, NVIDIA based cards are well supported by Linux but driver installation may be required.

The video rendering software will make use of the graphics card GPU by means of VDPAU (Video Decode and Presentation API for UNIX) package. This option (`libvdpau1`) should be installed if you have active, compatible, video drivers.

### 1.2.6 Hardware Timing

The system makes use of several standard Linux programs including image display programs, players, screen savers and camera display software as well as detached processes and multiple threading. Once initialized, the visual windows associated these are normally repositioned and resized, as needed, into the console area. As each of these operations takes a finite amount of time to load an initialize, the system has built in delays between the time when the service is requested (for example, a player) and the time when it
is repositioned and resized. You will notice these delays at times.

The load and initialization times for some features will depend upon the speed of your hardware and other loads concurrently running on your machine. Among these are:

1. Audio latency. Depending on the speed of your CPU and the number of processors, the audio may or may not be correctly aligned with video. This is due to the different processing for audio and video. There are settings in stream.script and VLC to compensate for this. Also note, the audio channels on some videos available on the Internet are not aligned correctly so compensation may not be advisable.

2. Camera initiation. When starting a camera in the system, a delay is taken while the camera initializes. If the camera does not fully initialize, subsequent control of the camera may not be successful. This may be remedied by increasing (or decreasing in the case of fast cameras) the delay in the code Automation.c.

3. Other windows. The system console incorporates windows that are created and maintained by other programs. As with cameras, there is a delay built in when these are started to give them time to fully initialize. These delays may need adjustment depending on your system.

Also, the basic internal clock for the system operates on a one second interval. As a result, some activities may not synchronize until the next clock cycle. The effect of these is minimal but may be visible.

If your hardware is slow, it may be necessary to increase the builtin delays to compensate.

### 1.3 Goals

Internet casting has been growing in popularity in recent years. While there are many commercial, and some open source, products available in the Windows ecosystem to support these activities, there are relatively few in Linux.

For actual casting, OBS (Open Broadcaster Software) is perhaps the best known. It runs on both Windows and Linux and has many sophisticated features to support casting and recording applications. Unfortunately, OBS requires considerable CPU resources, has complicated settings that do not always produce the best broadcast image for a given bit rate, and has more features than the typical user may need. Also, OBS is mainly a casting/recording tool. It lacks broadcast automation and playback facilities. These must be provided by other packages such as Mixxx, VLC, smPlayer, and so forth.

The attempt here is to develop an open source system that makes use of existing Linux resources, where available, that is compatible with OBS but has a simpler, less resource intensive profile, suitable for lower power machines, with support for:
1. high resolution full motion (24 fps) video
2. low streaming bit rates
3. built-in streaming encoder
4. broadcast and multi-media automation support.

The system makes use of existing software packages where ever possible. In some cases (pavumeter, pavucontrol, and guvcview), slightly modified versions of these standard packages are included in the distribution.

Existing software packages are started as independent processes. Since most of these take a period of time to initialize (for example guvcview can take several seconds to initialize a camera, depending on the camera), you will notice delays at times when first starting a service. These are due to built-in sleep commands that pause until the service has begun. The duration of these can be altered and, in some cases, depending on your hardware, may need to be increased.

### 1.4 Basic Design

The package consists of:

1. a control panel to schedule and automate the playing of local and Internet accessible media files;
2. an ffmpeg based encoder capable of sending a 24 frame per second (fps) full motion video to an RTMP (Real Time Messaging Protocol) media streaming server at a high level of compression;
3. PulseAudio loop-back configuration to merge and control audio send to the encoder.
4. Slide show, screen saver, camera, browser, and YouTube video controls.

### 1.5 Performance

In Linux, the most widely used audio and video casting system is OBS. OBS offers a number of features for both recording and casting but these can come at a price on lower power machines with limited graphics processing support, as is the case in Linux.

In OBS, multiple sources of video, audio, slides, video capture and so forth are organized into scenes. OBS captures the contents of these and allows the user arrange them into a window the contents of which are encoded and sent to the streaming server. Multiple scenes with differing layouts are possible. OBS supports a wide range of options for resolution frame rates, encoding and so forth.
A problem that arises, however, is that, on Linux systems, where video graphic card support in minimal, the real time processing to handle this can be overwhelming to even high end machines. Even simple casts can require more CPU power than many systems provide, especially if high compression rates are needed. Full motion video is often difficult to attain.

The concept used here is simpler. A fixed area of the user’s display monitor is reserved as the *casting area* or *canvas*. Anything placed in this area (players, video sources, browsers, terminal windows, etc.) will be encoded and transmitted. Audio is obtained and controlled through the PulseAudio system (a JACK version will be available soon).

While this approach is not as flexible as OBS, it is far simpler and requires, in most cases, less than half the CPU time as OBS to cast the same content at the same frame rate but at a one quarter the bit rate.

In the default configuration, full motion, 24 fps video at a resolution of 640x360 can be encoded and transmitted at bit rates less than 850K per second with no graphical video hardware requirements. In a comparison test on the same media file, to achieve full motion video at the same resolution, OBS used in excess of 2,500K bits per second and twice the CPU time.

On an AMD FX 6100 processor overclocked to a maximum of 4 gHz\(^1\), a full motion (24 fps) video at 640 by 350 resolution encoding to the iVlog.tv streaming host site used the following:

1. Overall 1 minute Linux load factor: 1.0
2. Bit rate: varied but never more than 835 k bits/second
3. Encoder (*ffmpeg*): between 47% and 53% usage of one CPU core
4. MPV used about 20% of one CPU core.
5. Pulse Audio used about 12% of one CPU core.

The system is suitable for laptops and portable applications where OBS will not run. A very limited standalone version can be made to run on a Raspberry Pi. In this mode, the control panel is not used and the casting area is smaller. Consequently, the CPU load is reduced.

### 1.6 Workspace Switching

If you switch workspaces while casting, the audio from a player in another workspace will continue to be streamed although the visible casting area will switch to the new workspace. See below in section 2.7 for details on the size of the areas being cast.

---

\(^1\) The CPU was in *on demand* mode such that the actual CPU frequency varied from 1.4 gHz to 4.0 gHz.
1.7 Implementation

The present implementation is on-going and has only been tested on **Linux Mint with Mate 18.3** using the **Mint X** (or similar) theme. It will work with other versions of Linux but adjustments may be required (see Section 8). The implementation depends upon the underlying window manager for many services and these services may or may not be available in other Linux distributions.

2 The Automation Console

2.1 Overview

The system consists of a casting console that controls the audio and video presented to the screen.

You have three streaming options:

1. Normal Mode: 640x350 resolution. The area streamed corresponds to the gray area shown in Figure 1. Normal Mode is selected when the **Panel** and **Full Screen** buttons are not highlighted. That is, when the text in these buttons is not red.

   In Normal Mode, players, camera feeds, screen savers, and slides are rendered within the gray casting area.

   In Normal Mode, the encoder only transmits the contents of the gray area. The remainder of the console and desktop are not visible in the stream being cast. In Normal Mode, the bit rate is about 600 kilo bits per second or less at 24 frames per second. The resulting cast is of good resolution with full motion video and suitable for most applications especially if the target is a tablet or phone.

2. Panel Mode: 1324x750 resolution. The area streamed corresponds to the entire area shown in Figure 1. Panel Mode is engaged by clicking the **Panel** button. The text in the Panel button will turn red to indicate that Panel mode is active.

   In Panel Mode, the encoder will transmit the contents of the entire console area. Video players, screen savers, and slide shows will initially display within the gray area while the camera feed will display in the space below the gray area as shown in Figure 3.

   In Panel Mode, you have the option of enlarging video players, slide shows, the camera feed and so forth to the full panel area by clicking the Fill buttons at the bottom of the console. These buttons have no effect in Normal Mode.
3. Full Screen Mode: 1920x1080 corresponding to an entire screen\(^2\). Full Screen Mode is enabled by clicking and highlighting (red) the Full Screen button. Clicking the Full Screen button turns off the Panel button if it was highlighted. Full Screen Mode is similar to Panel Mode except the entire desktop is streamed. This mode may be unsuitable for casts targeted at tablets or phones.

Although Panel and Full Screen modes are available, the intended primary casting area of the screen is the gray rectangle in Figure 1 which is 640 wide by 350 high. Visuals appearing in this area will appear in your transmission.

![Figure 1 Automation Console](image)

\(^2\) Screens with at least 1920x1080 are required at the moment. To use another resolution requires a modification of the source code.
Figure 2 Automation Console With VLC Video

Figure 3 Console in Panel Mode with Camera Feed
2.2 Audio / Video Players

Either the MPV or VLC audio/video player may be used. In these examples, the default VLC player is shown. VLC comes with a built-in equalizer and a built-in compressor as well as some other features that may make it a better choice.

2.3 Browsers

Both Chromium (for YouTube videos) and Chrome (for general browsing) are used. You need to install both on your system. Before using either Chromium of Chrome in this system, you must close all existing Chrome or Chromium windows.

2.4 Console Layout

Figure 1 shows the basic automation console. The casting area is the darker gray area in the upper right. Figure 2 shows the console with a video playing in the casting area using VLC^3.

The active areas of the console are as follows:

1. The internal system volume meters (upper left) L Mon and R Mon (instances of pavumeter, a program commonly part of Linux distributions^4) showing the sound levels being presented to the encoder.

2. The sound levels for the microphone input channel (upper left) L Mic and R Mic (pavumeter).

Both the Mic and Mon meters windows are separate windows that float above the control console. However, if you drag the control console to another location, they will reposition themselves to their original, relative positions.

3. The normal casting area is the rectangle in the upper right of the console - where videos, slide shows, the camera, and the screen saver will normally display. The area of the rectangle is 640 by 360. This is Normal Mode. It is engaged when Panel and Full Screen are both off (i.e., their button text is not red). Any image or video dragged to this area will be transmitted when the stream encoder is active.

---

^3 Choice of VLC or MPV is a compile-time option.
^4 The versions of pavumeter and pavucontrol provided in the distribution have been modified slightly from the versions normally included in Linux in order to reduce the size of the associated visible windows.
If there is no content to display (video, slide show, screen saver), the gray rectangle or one of the other static displays cards selected by the radio buttons shown below the gray casting area will be visible. You may select a replacement for the gray image image by clicking one of the radio buttons. Figure 4 shows this area with the Linux logo selected. You may add additional buttons by modifying the source code in Automation.c and the Glade file Automation.glade.

4. The master system audio control (lower left panel with tabs) shows, in Figure 2, the output level and volume control for the VLC player which is playing the video clip seen in the casting area. It is also used to control other aspects of the sound system.

Audio levels for all sources being sent to the encoder can be controlled from this window. When other sources are active, for example Skype or Chrome, their controls will appear here. The audio control panel is generated by a modified version of the standard Pulse Audio program pavucontrol provided with the distribution. This window floats above the panel and repositions accordingly if the main panel is moved.

---

5 The versions of pavumeter and pavucontrol provided in the distribution have been modified slightly from the versions normally included in Linux in order to reduce the size of the associated visible windows.
2.5 Console Placement
The console will be placed in the center of your desktop. This may be changed, at present, by adjusting:

#define DEFAULT_MW_HOR 300 // default placement
#define DEFAULT_MW_VER 100

in Automation.c.

2.6 Casting Panel Buttons and Boxes

2.6.1 File Select, Randomize, Text Entry and Play

File Select

There are two File Select buttons. Each opens a standard Linux GTK File Chooser window whose initial directory, the first time you use it, is:

$HOME/Desktop/Broadcast.

From there, you may navigate to other directories.

In the File Chooser, you may select a video or audio file to be played. The name of file selected will appear in the adjacent text box (shown with the shadow designation music file name).

Once you have selected a file, subsequent usage of the File Select button will cause the File Chooser to open in the same directory from which the selection was made however you may navigate to other directories if you wish. The File Chooser pop-up is not visible in the gray Normal mode casting area but would be in Panel or Full Screen mode. An example of the pop-up File Chooser for a File Select button is shown in Figure 5. Selection of a file does not initiate play of the file.
Randomize

The **Randomize** buttons will select a random media file from the directory from which the last file was chosen by the corresponding adjacent **File Select** button and place the file name into the corresponding text box. The **Randomize** button does not initiate play. The media file will play when the corresponding **Play** button is clicked. Use of the **Randomize** button is meaningful only if a directory has been identified by prior use of a corresponding **File Select** button.

Music File Text Entry Box

The box may receive entries from the File Select button or the Randomize button. You may also past entries into this box. File references must be full system file references (path from root). You may also place URLs of videos if the videos are directly playable. That is, they are **not** YouTube URLs as these are not directly playable by VLC or MPV.

Play

The Play buttons cause the named file, or URL to play in a window in the gray area.

Video Fill

The Video Fill button at the bottom will cause the video player to expand to fill
the entire panel in a manner similar to that shown for YouTube URLs (see below). Click the button a second time returns the video to the gray area. The Fill buttons only work in Panel or Full Screen modes.

2.6.2 YouTube Videos

YouTube URL, Text Entry Box and Play Button

The button labeled **YouTube URL** inserts the main YouTube URL into the YouTube text entry box to its right. If you click the corresponding **Play** button, the top page of YouTube will appear in a small Chromium window in the casting area. You may search and play YouTube videos from here.

The YouTube text entry area is also where URLs pointing to YouTube videos may be pasted (but not typed).

The YouTube video whose URL is placed here will be played in a small Chromium window that will appear in the gray casting area when the adjacent **Play** button is clicked in full panel side if the corresponding Fill button at the bottom is clicked and the system is in either Panel or Full Screen mode (see Figure 6).

**YouTube Button**

The YouTube button in the center turns on or off the display of the Chromium window containing a YouTube video.

**YouTube Fill Button**

The YouTube Fill button on the bottom of the panel will cause the Chromium window displaying a YouTube entry to expand to fill the entire panel as shown in Figure 6. The button is highlighted (red) when active. Clicking the button again will cause the YouTube window to return to the gray area and the panel to be restored.
2.6.3 Full Directory Select and VLC

Full Directory Select Button

Clicking the Full Directory Select button will bring up a directory chooser box with which you can navigate and select a directory.

VLC

The VLC button causes the music/video contents of the selected directory to play in random order. Only the VLC player is used, not MPV. If the VLC window has focus, the keystroke ‘n’ will advance to the next randomly selected file.

2.6.4 Playlist Select, Shuffle and Play

Playlist Select

The Playlist Select button functions in a manner similar to File Select. It is used to select a standard system playlist file (.m3u) such as may be produced by VLC or smPlayer. The full playlist file address will appear in the adjacent text box or you may paste the full path information of a playlist. Only playlists may be used, not regular media files. Regular media files will not play from this entry box.

Shuffle

Note: if VLC has focus, all keyboard shortcuts are available.
If you click the Shuffle button, the contents of the playlist file will be played in random order.

**Play**

If you click the Play button, the contents of the playlist will be played in consecutive order.

### 2.6.5 File Search

**Search Box**

The uppermost search box (identified by a magnifying glass icon), searches through the music library for files whose names satisfy the search expression entered into the search box. The search begins when the user hits enter. The search expression may be any regular expression accepted by `grep`. Up to 20 files found will be displayed. If you click on one, it’s path information will be entered into the box labeled Search Results. The music library searched is:

```
$HOME/Desktop/Broadcast
```

![Figure 7 File Search Example](image)

Figure 7 shows the results of a file search where the search term was *jackson*. The search result box is not visible in the cast in Normal mode but is visible in Panel or Full Screen mode.
Play
Clicking the adjacent Play button will cause the search results entry to play.

2.6.6 Program Search
The second search box searches through entries in the file `program.lst` from which the 12 program buttons are populated. The search expression is any expression accepted by `grep`. A results box will appear and the program buttons will rearrange such that the selected entry will become the upper rightmost button. The appearance of the search box is similar to that shown in Figure 7.

2.6.7 ffmpegV and ffmpegI
The `ffmpegV` and `ffmpegI` buttons toggle the stream encoder on or off. When the encoder is on, the stream bit rate will be shown in the area labeled (at this moment) `stream inactive`. This buttons invoke the bash script file `stream.script` which you must edit to insert your streaming host’s URL and your casting key. The encoder will not function until you do so. As currently configured, `ffmpegV` turns on the section of code in `stream.script` presently configured for VaucghLive while `ffmpegI` turns on the section presently configured for iVlog. You may change these to target other streaming servers.

2.6.8 OBS Button
The OBS button starts the OBS studio. You must configure OBS separately (see below). Once started, you must terminate OBS from the OBS window itself.

2.6.9 Panel Button
The Panel button tells the system that you will be casting the panel area (rather than full screen or the gray casting area). The Fill buttons at the bottom of the panel will not work if Panel (or Full Screen) is not in effect. They require Panel mode or Full Screen mode. If you are already casting (with ffmpeg), neither the Panel nor the Full Screen button will not respond if clicked. They are locked during streaming.

2.6.10 Full Screen Button
The Full Screen button tells the encoder that you will be casting the entire screen (1920 x 1080). It will not respond if you are already casting. The Fill buttons at the bottom of the panel will not operate if you are not in Panel or Full Screen mode.
2.6.11 Normal Resolution Mode

If both Panel and Full Screen are not highlighted, casting will be confined to the contents of the small gray area.

2.6.12 Program Buttons

The three columns of large buttons in the center of the control panel are the program buttons. Each button shows the name of a file read from the file program.lst. The file names are shown with path information omitted (the program.lst file must contain path information - see below).

Each file name that appears has been processed so that: (1) special characters removed, (2) the first character of each word following a blank capitalized, and (3) all other text reduced to lower case. The numbers in the parentheses following the file name are (1) the number of seconds of play time for the file and (2) how many times the file has been played during this session.

Left clicking on a button causes the named file to play.

Right clicking on a button causes the display of buttons to reorganize such that the button right-clicked moves into the first position (upper leftmost position).

2.6.13 Seq Program Button

The Seq Program button will begin to play the media files whose names appear in the program buttons. The files will be played in the order (sequential) in which they appear. The first file to be played will be the one designated by the upper leftmost button. As each file is played, the buttons shift. The currently playing file is always in the upper left button. A media file may be moved to the first (upper left) button by right clicking on it. The files whose names appear in the buttons are loaded from program.lst which may be organized into a program play list.

2.6.14 Rand Program Button

The Rand Program will randomly select media from the program.lst file. The program buttons will be rearranged such that the currently playing media file is in the upper left button.

2.6.15 Stop Players Button

The Stop Players button halts program play and closes the player.
2.6.16 Pause Button

The Pause button will cause the currently playing sequential or random program to pause at the end of the current file and wait until the button is clicked again. This allows for announcements between files.

2.6.17 Origin Button

The Origin button returns the program buttons to the beginning of the program.lst file.

2.6.18 Next Button

The Next button, when in Seq Program or Rand Program mode, causes play of the current media file to end the current file and to advance to the next program file.

2.6.19 Prog < Button

This button shifts the program button page one page to the left.

2.6.20 Prog > Button

This button advances the program button page one page to the right.

2.6.21 Skype

The Skype button starts the Skype program (skypeforlinux) and places the Skype window in the upper right corner of the screen. The Skype window may be dragged to the casting area if desired. Audio from the Skype app is available to the encoder and can be controlled from the audio panel in the lower left of the console.

2.6.22 Saver Button

Saver displays the screen saver in the casting area. The default screen saver is the morph3d screen saver normally distributed with Linux. If morph3d is not present, this will not work unless another saver has been compiled into the code. The screensavers are assumed to be in /usr/lib/xscreensaver/ and the section of Automation.c that controls the screen saver is at the symbols SS_LIB and SS_NAME.

2.6.23 Slides Button

Initiates the slide show in the casting area. The slide show is derived from the directory $HOME/Desktop/Broadcast/Slides640. The images should be edited to be no wider than 640 pixels.
2.6.24 Video Button

Causes any currently running player (Chrome app or MPV) to appear in the gray casting area or become invisible (toggle).

2.6.25 Cam0 Button

Toggles the camera on or off and places the video stream into the casting area (when not in Panel or Full Screen mode) or in the area below the gray casting area when in Full Screen or Panel mode.

The system assumes a camera resolution of 1280x720 (or similar ratio - this can usually be adjusted and set by guvcview) The camera is assumed to be named /dev/video0.

This button invokes a modified version of guvcview (provided in the distribution, see below). If you have a second camera, you may manually open it and drag its display to the casting area as desired. A new camera should be opened first by guvcview independent of this program to create default settings.

Due to variations in hardware speeds, the delay factor for the camera may need to be adjusted. If the camera does not position itself correctly, this is a sign that the delay factor needs to be increased.

2.6.26 Cam Inset Button

The Cam Inset Button causes the camera image, if the camera is running, to appear as in inset in the casting area in Normal Mode as shown in Figure 8. It has no effect in Panel of Full Screen mode as the camera, if running, is always displayed in these modes.

The placement and size of the camera inset image is controlled by settings in automation.config.
2.6.27 **YouTube Button**

Toggles the visibility of a YouTube window.

2.6.28 **Hide All Button**

Hides all casting area content.

2.6.29 **Chrome Button**

Toggles visibility of a Chrome browser

2.6.30 **Casting Area Background Radio Buttons**

The radio buttons beneath the casting area control the still card display in the casting area. Other options are test patterns, color bars, Linux logo, and a road scene.

2.7 **Casting Modes**

There are three casting modes corresponding to three resolutions. In each mode, if you switch to a different workspace, the area of the new workspace that corresponds to the casting area from the previous workspace is the area transmitted. There is no loss of audio
due to switching workspaces as can be the case in OBS.

2.7.1 Low Resolution Mode
In low mode (where neither Panel nor Full Screen are selected), the mode in which the console initializes, only items placed in the shaded gray area on the upper right of the console will be transmitted when ffmpeg is running. The resolution of the cast will be 640 wide by 360 high. When ffmpeg is running, you must not move the console window.

The ffmpeg will lock onto the gray casting area coordinates. Any attempts to move the console while ffmpeg is running will fail.

In low resolution mode, typical bit rates at 24 frames per second are about 400 k bits per second.

2.7.2 Panel Resolution Mode
In panel cast mode, the entire panel will be transmitted when ffmpeg is on. The resolution of the cast is 1324 wide by 750 high. In panel mode, the console window will retreat to the upper left of the screen and attempts to move it will fail.

In panel mode, typical bit rates at 24 frames per second are around 800 k bits per second.

2.7.3 Full Screen Resolution Mode
In full screen mode, the entire desktop is transmitted when ffmpeg is on. The resolution of the cast is 1920 wide by 1080 high. If your screen is too small for these dimensions, do not use this mode or, recompile Automation.c with the dimensions of your screen. In this mode, you can move anything anywhere.

In full screen mode, bit rates at 24 fps are typically about 1200 k bits/second.

2.8 Program Buttons
The 12 visible program buttons contain entries taken from the file program.lst which is loaded when the system is started.

A left click on one of these buttons will cause the file referenced to play. A right click will cause the buttons to shift such that the button right-clicked will appear in the first position (upper leftmost).

The Prog< and Prog> buttons will advance (or retreat) to the next twelve entries.

If the Seq Program is clicked, the entries referenced in the program buttons will play one after another (sequentially) beginning with the entry in the upper left button (see note above about right-clicking a button).
The maximum number of program entries is determined by the compile time defined symbol `SOUNDER_MAX`. The default value is 4096.

### 2.9 Search Boxes

There are two search boxes. One searches for files in the music library and the other searches entries in `program.lst`. Figure 9 shows an example of a music library search. The `program.lst` file search is similar in appearance.

#### 2.9.1 Music Library Search

All file names in or beneath `$HOME/Desktop/Broadcast` will be searched. The search term may be any regular expression legal to `grep`. Typical search entries, however, will consist of a single word. Case is ignored.

A list of files satisfying the search criteria will be displayed. The file selected will be entered in the box labeled `music file search result`. Clicking on the adjacent `Play` button will cause the file to be played.

#### 2.9.2 Program Search

The other search box, labeled `Search program entries`, searches the entries in `program.lst`.

---

7 `grep` is a standard Linux command line search program that accepts regular expressions.

28
Files matching the search expression are displayed. If a file is selected, the buttons are rearranged such that the selected file is in the first button position (upper most left). The search expression may be any regular expression accepted by `grep`.

## 2.10 Screen Saver Mode

It is not uncommon in casts, when there is no meaningful video available, for example, when an audio only file is playing, to show a still card, and animation or a pre-formatted slide show.

The screen saver mode used here consists of a display of animated patterns taken from the `xscreensaver` package which is part of many Linux distributions. The one shown in these examples is the `noof` saver. There are many others. Selection of which screensaver to use is presently a compiled option but selections buttons should be available soon.

The current default screen saver is `morph3d`. The full list of available screen savers is visible in the directory:

```
/usr/lib/xscreensaver
```

and a sample of these can be seen in:

```
System | Preferences | Look and Feel | Screensaver
```

You may change the installed screen saver by modifying the defined symbols

```c
#define SS_LIB "/usr/lib/xscreensaver/morph3d &"
#define SS_NAME "morph3d"
```

in `Automation.c` and recompiling.

When selecting a screen saver, you should check the amount of video it generates and the amount of CPU time it uses. Some other savers that might be of interest are `deco`, `fuzzyflakes`, `gears`, and `glsnake`.

Generally speaking, screen savers generate a great deal of video and thus may increase bandwidth. They should be used sparingly.

---

8 See the symbols SS_LIB and SS_NAME in Automation.c
2.11 Slide Show Mode

The slide show is a display of images that part of the cast. The slide show presenter is *Phototonic*.

You should run and set *Phototonic*’s preferences before you use it here. Ideally, slide show should be set to *random*, time should be set to 10 seconds per slide, and not to display file name in the slide.

The slide show defaults to show images from the directory:

```
$HOME/Broadcast/Slides640/
```

Due to the way Phototonics works, an initial image is required. The default is:

```
image.jpg
```

So, you should ensure that the directory has an initial image named *image.jpg*. 
2.12 Skype View

In Skype view, the `skypeforlinux` program is executed and the window for Skype is placed on the right side of the display monitor. Skype audio is available to the audio control panel and video images may be cast if you move video portion of the Skype window into the casting frame. You must first install `skypeforlinux` if you wish to use this option.
2.13 YouTube View

When the Play button adjacent to a YouTube URL is clicked, a small Chromium window app is started and the URL from the panel is displayed. If this URL is a video, the video will play and the audio will become available to the audio control panel. If the main YouTube URL is inserted into the text box by the YouTube button and the Play button clicked, the YouTube front page will appear in the Chromium app window from which you can search and play videos.
The system can be used as a set of source windows for OBS where OBS can be used for recording or stream encoding. In this view, you have the option of only broadcasting the casting area or you may cast the main panel, the microphone and system volume levels, the audio control panel and the players which are each separate windows that may capture with OBS.

If you click the OBS on/off button, it will start OBS as shown in Figure 14 (or terminate OBS if it is running). The main console window will reposition to the upper left hand corner and OBS will appear in the location where you last positioned it.

Ideally, you will want to minimize the size of the OBS console and eliminate unneeded elements. In

View | Docks

you can determine whether scene transitions, scenes, controls, sources, etc appear in the OBS console. The fewer present, the smaller you may make the OBS console. Note: after you have resized the OBS console, you should exit and restart to see the full effect.

Tests indicate that the OBS encoder uses in excess of twice the CPU time and requires approximately four times the band width to produce a cast of similar quality to the builtin
ffmpeg encoder.

### 2.14.1 OBS Settings

The OBS configuration shown in Figure 14 is based on an OBS screen capture source where that only broadcasts the casting area of the console.

When you click OBS the console will reposition to the upper left hand corner of your display.

Since the casting area of the console displays several independent windows, you may want to use the OBS screen capture source.

To do this, you need to set the screen capture transform (right click on screen capture and then click transform and then edit transform) to crop only to the casting area:

1. crop left is 700
2. crop right is 590
3. crop top is 28
4. crop bottom is 684

The cropped area corresponds to the casting area of the console.

Other OBS settings recommended for use with the casting console are:

Settings | Video | Output Resolution

set to:

640 by 360.

The output settings:

Settings | Output | Advanced

are:

Rate Control: CBR
Bitrate 2500
CPU: medium
Before using OBS, you must also set your streaming server’s URL and you casting key in the OBS settings.

### 2.15 Camera View

Camera view assumes a video camera of resolution 1280x720 or similar proportions. In camera view, you drag and resize one or more camera windows into the casting area. The contents of the casting window will be presented to the encoder. Other elements may be in the casting area at the same time.

The camera view uses a modified version of the program *guvcview* which is normally distributed with many versions of Linux. The modified source code and a compiled instance are in the Automation directory. The modifications involved the title bar which, by default, shows the frame rate. The modified version shows */dev/video0* which makes the window easier to control.

If you need to adjust your cameras settings or initialize it for the first time (many cameras need to be initialized at least once), use the Linux distribution provided *guvcview*. The command line:

```
guvvcview --gui=gtk3
```

may be needed to start the camera control panel.
Since cameras can take a few seconds to initialize, there are delays built into the camera initialization code. Pauses during camera initialization are normal.

![Figure 15 Camera View](image)

### 3 Encoder

The built-in encoder is `ffmpeg`. It is invoked from the `bash` script file `stream.script`. It is normally invoked by `Automation-bin`.

It may, however, be invoked directly. If invoked directly, you should provide three command line parameters:

```
stream.script vert-offset hor-offset widthxheight
```

where:

- `vert-offset` is the vertical offset of the upper left hand corner of the casting area from the top edge of the screen.
- `hor-offset` is the horizontal offset of the upper left hand corner of the casting area from the left hand side of the screen.
- `widthxheight` (example: 640x380) is the width and height of the casting area in pixels. Note there must be a letter `x` between the two numbers and no blanks are permitted. The numbers should be even.
If you wish to use the loop back audio, you need to invoke `loopback.script` prior to running `stream.script`. See `runAutomation.script` for details on how to kill the loop back after you have finished.

## 4 Standalone Mode

The encoder and audio system can be run in standalone mode without the use of the automation panel. In this mode, the user is responsible for initiating and moving media files into the casting area. This mode is useful on very limited machines such as Raspberry Pi where the casting area is a small area of the home display thus reducing CPU usage.

## 5 Installation

### 5.1 Linux Base

The system, as of this writing, has been tested on **Linux Mint 18.3 with Mate** and the default **Mint X** desktop theme. This code may work on other distribution and window managers but it has not been tested as of this time.

### 5.2 Desktop Theme

GTK desktop themes influence the manner in which widgets (buttons, labels, lists, etc.) are displayed. The widgets in this project are tightly grouped to save screen space. If the default theme expands the padding or spacing of these, they may overlap or otherwise not display correctly. The theme **Mint X** is the default at present with **Linux Mint 18.3 with MATE**. Other themes should work as well but you should check their effects on not only this software but others as well.

You can set the theme by going to:

```
System | Preferences | Look and Feel | Appearance
```

In the pop-up box, select Customize, and under Controls, select **Mint X** or similar.

Usage of incompatible control themes may cause erratic layouts.

### 5.3 Installation

1. The distribution comes with binary executables which are compatible with **Linux Mint Mate 18.3** and, probably, related Ubuntu based systems if auxiliary software is installed. The script file `install.script` consists of several `apt-get` commands that, if executed as root, install the needed code if it is not already present.
2. Unzip the distribution onto your **Desktop** directory. Do not attempt to install it elsewhere. It will not work.

3. Create (as root) the directory */usr/local/share/pavucontrol*

4. Copy the file in the distro named **pavucontrol.glade** to the above.

5. Make it world readable (**chmod a+r pavucontrol.glade**) 

6. Be sure the directory path to it is world accessible (x permission for directories but this should be the default) 

7. Install the programs (from synaptic) **mpv, vlc, and chromium**

8. Create a directory on your **Desktop** named **Broadcast**

9. Put some video or audio files into this directory (or subdirectories of same).

10. Make your **PATH** contain the current directory. Add the following, at the end, to **.bashrc**

    ```bash
    PATH=$PATH:$HOME/bin:.:/bin:/sbin:
    ```

    Close & reopen your terminal window for the above to take effect.

11. See Section 6.4 for information on what directories are required and their contents.

## 6 Running the Console and Casting

### 6.1 Environment

You should close other windows you may have running on your system. Possible window name overlaps may result in the console misplacing dependent windows.

For example, if you have an instance of Chrome or Chromium running, these may interfere with the instances that may be started by the console.

### 6.2 Set Up Program.lst

Replace the contents of the file **program.lst** with your music file names. This is the file that will be used to load the buttons with music. The file must consist of full file references, one per line, for the music you want to have appear in the buttons. For example:

```
/home/you/Desktop/Broadcast/RockMusic1/processed.full/VenturaHighway.mp4
```

---

9 .bashrc is in your home directory.
The above should be an absolute file address on your system where you is your user id. You can get a list of absolute addresses for a directory of media files on your system by the following:

1. Open a **caja** file explorer window to show the file names in *compact* format (see the edit preferences option).
2. In the window displaying the files in compact format, type ^a (control-a) followed by ^c (control-c). This means *all*, and *copy*.
3. Open the text editor (**xed**) from Applications | Accessories
4. In the text editor window, type ^v (paste). The edit window will now contain the file names. Remove any references to non-media files.
5. Save this as *program.lst* in the **Automation** directory. The files in this list, in the order they appear, will be shown in the playlist buttons of the control console.

If you successfully built *program.lst*, you will now see the file names listed in the buttons. Click on one and it should begin to play. If there is no *program.lst* file, the buttons will be empty.

The maximum number of entries in *program.lst* is determined by the compile time defined symbol **SOUNDER_MAX** which is set to 4096 in the distribution.

### 6.3 Server URL and Casting Key

To begin a cast, first modify the **SERVER** and **KEY** variables in *stream.script* to point to *ivlog* or *vaughnlive* or what ever other service you intend to use that supports the *rtmp* (Real Time Messaging Protocol). This information should be available at the streaming server host site.

### 6.4 Directories

The system has been configured to assume that certain files will be located in fixed locations. In future versions, some of these will be relaxed but, for the moment, these locations are **required**.

1. The system assumes that the environment variable **$HOME** contains the location of your home directory (this is normally the case in most Linux distributions).
2. The system assumes that **$HOME/Desktop/Broadcast** is the directory in which your media files are located. The directory may contain sub-directories. Symbolic links are permitted but the search function does not follow them at present.
3. The system assumes that **$HOME/Desktop/Automation** is the address of the directory containing the systems software.
4. The system assumes that $HOME/Desktop/Automation/fifo is the location of one of the system pipes used for internal communication. This file will be automatically created and deleted. If, however, it does not get deleted automatically, you will get an error message if you try to copy it (or the containing directory). If the console is not running, you may safely delete this file.

5. The system assumes that $HOME/Desktop/Automation/stream is the location of one of the system pipes used for internal communication. This file will be automatically created and deleted. If, however, it does not get deleted automatically, you will get an error message if you try to copy it (or the containing directory). If the console is not running, you may safely delete this file.

6. The slide show default directory is $HOME/Broadcast/Slides640. The slide show requires an initial image file by the name of image.jpg to be located in this directory.

6.5 Executing the Console

You initiate the system with the command\textsuperscript{10}:

\texttt{./runAutomation.script}

This will cause the main console to initiate and start the floating windows.

As the system initiates, several small windows will open, and, ultimately, reposition themselves on the screen. This can take a few seconds as there are delays built into the startup procedure to allow these windows to initialize. On some slower systems, it is possible (but not likely) that the builtin delays may not be sufficient. If this is the case, they should be increased in the main code module (Automation.c).

You will also see messages in the terminal window that initiated the system. Some will be minor error messages and this is normally not a problem. Many will be from other software packages such as Phototonics and guvcview. If it crashes, however, there were problems.

Once the console has finished initialization, you may initiate casting by clicking the Stream on/off toggle button. This will cause the script file stream.script to execute and attempt to make contact with the server.

When the stream is running, do not move the main window as the streaming encoder has locked onto the casting rectangle. If the casting window is moved, the encoder will no longer see its contents.

When contact is established, the Stream Inactive text above the ffmpeg on/off toggle buttons

\textsuperscript{10} Note: the ./ is not required if you included the current directory in PATH as shown above.
will begin to display the encoder’s bit rate (the rate at which it is sending data to the server).

You may begin playing media files at any time.

You may terminate the console (and all floating windows) by clicking the X box in the upper right corner of the main window.

6.6 Selecting a Mode

You need to select a mode and resolution for your cast. The default mode is set in automation.config (see section 9.1). The default will be either Panel Mode or Normal Mode. If Panel Mode is the default, the Panel Mode button will be highlighted. See section 2.7 for details.

6.7 Enabling Slides, Saver, and Camera

Because they take time to initialize, if you intend to use slide shows, the camera and/or the screen saver, you should initiate them before you begin your cast. See section 6.4 for details on slide shows. The determination as to which screen saver will be used depends on a setting in automation.config (see section 9.1).

Each option, savers, slides and camera, even if not visible, will add to overall CPU load. The camera and screen saver use the most CPU time while the slide show (Phototonic) uses the least. You may terminate any of these during your cast by right-clicking on the associated button in the system launcher panel and selecting close.

6.8 Starting the Stream

Click the appropriate ffmpeg button to start the stream. Clicking it again halts the stream. Once the stream has begun, the console may not be moved nor may you switch mode (normal, panel, full screen).

6.9 Playing Audio and/or Video Files

There are several ways in which to play a file:

6.9.1 The Rand Program Button

The Rand Program button will randomly select and play files found in the program list buttons (loaded from program.lst). It will continue doing so until you click the Stop Players button.

If you click the Pause button, the player will pause between selections and wait for you to click Pause again.

If you want to skip to the next randomly selected file, click the Next button.
Note: in Playlist or Full Directory modes, the Next button does not work. In those cases, you give the player focus and type the letter $n$.

### 6.9.2 The Seq Program Button

The Seq Program button will cause the files in the program buttons to play in consecutive order beginning with the file listed in the upper leftmost button. A file moves to the upper leftmost button by right clicking on its button.

If you click the Pause button, the player will pause between selections and wait for you to click Pause again.

If you want to skip to the next file, click the Next button.

Note: in Playlist or Full Directory modes, the Next button does not work. In those cases, you give the player focus and type the letter $n$.

### 6.9.3 File Select / Randomize / Play Buttons

There are two rows beginning with File Select buttons. The are used to play individually selected files. The File Select button allows you to navigate to a file to be played. The Play button plays this file. The Randomize button will randomly select another file from the same directory as the file selected by the File Select button (do not use it until you have selected at least one file and, thus, a directory).

### 6.9.4 Full Directory Select / Play

This button is used to identify a directory and the Play button will cause the files in the directory to be randomly selected and played until the Stop Players button is clicked.

In this mode, the Next and Pause buttons do not work. To advance to the next file, give the player focus and type the letter $n$. You can give the player focus by clicking on its header bar or by clicking on its button in the launcher panel. Do not click in the play area as this will pause the player.

### 6.9.5 Playlist Select / Shuffle / Play Buttons

The Playlist Select button is used to locate a $m3u$ playlist file. The Shuffle button will cause the contents of this file to be randomly played. The Play button will cause the contents to be played consecutively. Play will end when you click the Stop Players button. In this mode, the Next and Pause buttons are not functional. Use the $n$ keystroke as noted above.

### 6.9.6 Insert Youtube URL / Play Buttons

The Insert Youtube URL button will insert into the adjacent text box the main URL for Youtube.
Alternatively, you may paste the URL of a specific Youtube video. Clinking the Play button brings up an app version of Chromium that will display and play the URL selected. If you use the main Youtube URL, you will need to use the Youtube search function to locate the video you want to play. Play will end when the Stop Players button is clicked. Next and Pause are not operative in this mode.

6.9.7 File Search and Play Button
The File search box will search the Broadcast directory (and sub directories) for files whose names match the search criteria. The list of located files will be presented. Selecting one will cause its name to appear in the adjacent play box and clicking Play will cause it to play. The search criteria are not case sensitive and may include only a fragment of the name of the file. Full grep compatible regular expressions are also accepted.

6.9.8 Program Entry Search
The program entry search box will search the files named in program.lst and cause the buttons to rearrange such that the file found will be in the upper leftmost box. Clicking the button containing the name will cause it to play.

6.9.9 Program Buttons
If you click on a program button, its file will play and its button will move to the upper leftmost position. You may navigate through the program buttons with the Prog < and Prog > buttons. Note: the first time a page of buttons is displayed, the system queries each file on the page for play time. This can result in a slight delay. Once the time have been loaded, they will not be loaded again for the duration of the current session.

7 Installation & Compilation

7.1.1 Required Software
The system makes use of a number of freely available, open-source software packages, many of which may already be installed and no further action is required. The system also assumes that the standard Linux Mint screen saver collection (/usr/lib/xscreensaver) is installed. The system uses the morph3d screen saver (this is a setting in Automation.c)

The following items are required:

1. ffmpeg
2. libgtk-3-dev
3. mpv video player
4. pavucontrol (a modified version provided with distro)
5. pavumeter (a modified version provided with distro)
6. PulseAudio (included in Mint 18.3)
7. gcc/g++ compiler and libraries
8. guvcview (a modified version provided with distro)
9. JACK Audio (optional if you want to use JACK instead of Pulse)
10. libpulse-dev
11. libgtkmm-3.0-dev
12. libgtk-3-0
13. libcanberra-gtk3-dev
14. libcanberra-dev
15. gtkmm
16. wmctrl
17. pulseaudio-utils
18. at-spi2-core
19. glade
20. Phototonic
21. chromium-browser
22. VLC
23. google-chrome-stable

These are available in the Synaptic package manager or you may automate the installation with the *install.script* file.

### 7.1.2 Installation Script

At this time a traditional `configure / make` install is not available. Instead, the process of installing missing items from the list above is automated in the script file *install.script*. This file consists of a sequence of instances of `apt-get` that will install any items from the above that are missing. You need to run it as root.

    sudo ./install.script

This script will generate a large number of messages (this is normal) and may take some time.
to run. You must have a functional Internet connection to use it.

You should now follow the instructions given in section 5.3 on page 37 above.

### 7.1.3 Compiling the Code

The distribution contains executable files (if you make them executable). However, once you have installed the required software, you may wish to compile the system which can be done with the command:

```bash
./compileAutomation.script
```

### 7.1.4 Modified Open Source Code

Included with the distribution are compiled versions of `pavumeter`, `pavucontrol` and `guvrcview`. These modifications to the standard versions make the windows that they generate smaller and easier to control.

To recompile the modified code (probably not necessary), descend into the appropriate sub-directory of `Automation`, directory and then, as root execute:

```
bootstrap.sh
make
make install
```

The executable will be in the `src` sub-directory (for `pavumeter` and `pavucontrol`) or `guvrcview` (for `guvrcview`). Copy the executables to the main Automation directory. The `configure` procedure for these may identify additional software requirents.

For `pavumeter` and `pavucontrol`, follow the procedure outlined in section 5.3 on page 37 above

On some systems, it may be necessary to run `pavucontrol` (Volume Control) once in order to initialize the pulse settings in your `~/.config/pulse` should no settings be present.

### 8 GTK Configuration

#### 8.1 Overview

The main console display shown in the examples is produced by the Gnome GTK+ toolkit. The widgets produced by GTK interact with the GTK window theme and settings on your desktop to render a visible result.

The package was developed with Linux Mint 18.3 with Mate using the default theme and font settings. It appears to work well with Mint XFCE and Cinnamon.

However, due to differences in themes, window managers, and default fonts and font sizes
used in other distributions, some tuning may be needed.

There are three ways to tune the visual display:

1. Adjust your window manager theme and font.
2. Modify the file `Automation.glade`
3. Modify the file `automation.css` (requires re-compilation). In particular, change the values of:

   font-size: 10px;

   to higher or lower and restart.

It is anticipated that there will be a configuration script that will automatically adjust these settings for the major Linux distributions in the near future. At present, however, Linux Mint 18.3 is the best option.

### 8.2 Automation.glade

The layout of the buttons and boxes is controlled by the file `Automation.glade` which is built and maintained by the `glade` tool (installed by `install.script`).

Within limits, adjustments to the appearance of the console may be made by altering this file, saving the result, and, in some cases, recompiling the code (usually not needed if only `Automation.glade` is changed). The Glade tool displays `Automation.glade` as shown in Figure 16.

### 8.3 Automation.css

The file `automation.css` contains code that overrides some of the default CSS settings used by Glade and GTK. These include setting the size of the buttons (default buttons are usually much larger), color of the button text, and the size of the button font.

On some versions of Linux, the default font size is either too large or too small. You may compensate for this in `Automation.c`

Other distribution differences, such as added window panels (Ubuntu, for example), may require modification of settings in `Automation.c` which control the placement of dependent windows in the console.

If you modify this file, you need to recompile `Automation.c`
For the most part, options at present are set in Automation.c as variables and defined symbols.

Options are also set in stream.script and loopback.script.

9.1 Automation.config

The file automation.config contains run-time configuration options that are user settable without recompilation.

The file format is very specific and sensitive to errors. The format rules are:

1. A comment begins with a pound sign (#) in column 1. The remainder of the line is ignored.

2. A parameter line consists of:
   (a) a parameter name beginning in column one.
   (b) the name must be typed exactly - it is case sensitive - with no embedded blanks
   (c) one (1) blank
If an option appears more than once in `automation.config`, the last instance is the one used. The version of `automation.config` provided, for example purposes, has alternative several duplicate settings. Only the last is used.

The following lists the options available at present and their default values.

9.1.1 Console Placement

DEFAULT_MW_HOR 300
DEFAULT_MW_VER 100

The location of the upper left corner of the console. The default is to center the console on the screen.

9.1.2 Default View

panelView 0

If zero (0) - the default - the console will begin in Normal Mode. If one (1), it will begin in Panel Mode. No other values are accepted.

9.1.3 Camera Delay

CAM0_DELAY 2

The number of seconds to wait until the camera initializes. At present, only on camera is supported (CAM0)

9.1.4 Camera Inset

# small insert in lower right corner

CAM_INSET_WID 160
CAM_INSET_HGT 90
CAM_INSET_HOR 476
CAM_INSET_VER 267

The above are, respectively, the width, height, and coordinates of the upper left corner of the inset. The numbers are relative to the casting area of the screen. The numbers shown position the camera in the lower right as seen in Figure 8.

9.1.5 Screensaver

SS_LIB /usr/lib/xscreensaver/flyingtoasters
SS_NAME flyingtoasters
The location (SS_LIB) and name (SS_NAME) of the screen saver.

### 9.1.6 Number of Program Buttons

Number of program buttons. This number must be larger than the number of lines in the file `program.lst` or not all lines will be loaded.

```
SOUNDER_MAX 4096
```

### 9.1.7 Play Time Load

Whether to load the play times of entries in `program.lst`. Loading the play time can be slow as each file must be interrogated. There is also a compile time option to decide if the play times should be loaded (`PLAY_TIME_LATE`) when the program starts (long delay for large `program.lst` files) or when the buttons are loaded.

```
PLAY_TIME_LOAD 0
```

### 10 Script Files

The distribution comes with several BASH script files. These all have a `.script` file extension. They are used to install, compile and run the programs. These all need to be user executable (should be the default). To make a file executable, in the Automation directory, type:

```
chmod u+x *.script
```

#### 10.1 install.script

This file installs Linux system software needed for the other programs. It consists of several `apt-get install` lines. It needs to be run as `root`. At the end, it compiles and runs the distribution.

#### 10.2 compileAutomation.script

This script compiles the system code. The system does not use the `configure / make / make install` sequence.

#### 10.3 loopback.script

This file sets up a Pulse Audio loop back to capture audio for streaming. This file is invoked by `runAutomation.script`.

If you are using OBS for streaming, it is not necessary to execute this file. The audio sources can be configured directly in OBS and you should remove the reference to it from `runAutomation.script`.

49
If you are using the built-in streaming facility, you need to execute this file prior to running the stand-alone streaming script file (*stream.script*).

The main execution script, *runAutomation.script*, automatically executes this file.

Audio normally consists of two sources: input from microphone/line-in, and the output of players (such as VLC). However, the encoder, *ffmpeg* is set to take audio from only one source.

One way to merge both audio sources is for *ffmpeg* to take the audio at the ALSA (advanced Linux sound architecture) level. On Linux, ALSA operates at the kernel level and controls system sound card(s). PulseAudio is an audio server that accepts audio from one or more sources. See Figure 17.
Figure 17 Linux Audio
In tests, however, this resulted in spurious noises in the audio stream that were not present when Pulse was used.

Consequently, Pulse was selected for the default audio server\(^\text{11}\).

In order for \texttt{ffmpeg} to be able to read both audio sources when using Pulse, it was necessary to route both sources to a single, internal audio sink which then becomes the sole source for \texttt{ffmpeg}.

The script \texttt{loopback.script} sets up the internal audio sink and routes the audio from both the computer’s microphone input, and output from any players (MPV, VLC, Chrome, etc.) to this sink. The output of this sink (\texttt{mywiretap.monitor}) is then becomes the input to the \texttt{ffmpeg} encoder (see \texttt{stream.script}) . Otherwise, the encoder would only receive audio from the microphone or the players, but not both.

The script file \texttt{loopback.script} attempts to identify the microphone and internal audio sinks. This code is a bit unstable so you need to give it some help:

To wit, if your system has multiple audio sources, such as a web cam or the audio channel on a video card HDMI port, you should disable these as they cause \texttt{loopback.script} to confuse them with the main audio card.

You can disable an audio device by right clicking on the speaker icon which should be on your panel and clicking on sound preferences then selecting the \textit{Hardware} tab. The devices will remain disabled until you turn them back on again.

\section*{10.4 \texttt{stream.script}}

The file \texttt{stream.script} contains the encoder commands. It is automatically executed by \texttt{runAutomation.script}.

If you use OBS to encode your stream, you should not execute this file.

If you use this file to encode your stream, you first will need to edit it to include the URL of the host server to which you are casting and your casting key (provided by the casting host).

\section*{10.5 \texttt{runAutomation.script}}

The file \texttt{runAutomation.script} executes \texttt{loopback.script} and runs the automation system. The automation system will invoke \texttt{stream.script} only if you click the \textit{stream on/off} button.

\footnote{\textit{Note:} JACK may be used in place of Pulse but JACK is significantly more complicated to install and configure and browsers do not, at this point in time, provide JACK output.}
11 FAQ

11.1 No Sound
Check that one or more of the volume controls has not been set to silent. If you, for example, set the output of *mpv* or *vlc* to silent, it will remain at silent until you change it - even if you reboot!

11.2 No Sound, Yet Again
Check that sound cards not being used are turned off (see above) and reboot. The file *loopback.script* attempts to determine which card is your main audio card and sometimes gets confused.

11.3 No Sound, Even More
Check your Sound Preferences to see if you have Input set to Built-in Audio Analog Stereo (the usual case for most systems) as shown in Figure 18.

![Figure 18 Sound Preferences](image-url)
11.4 No Sound, Not Done Yet

See Section 10.3 on the `loopback.script` file.
Alphabetical Index

1324 wide by 750 high.................................................................27
1324x750......................................................................................12
1920 wide by 1080 high.................................................................27
1920x1080.....................................................................................13
640 wide by 360 high.................................................................27
640x350.........................................................................................12
ACK...............................................................................................11
ALSA...............................................................................................50
apt-get.........................................................................................44
apt-get.........................................................................................37
Audio latency.................................................................9
Automation..............................................................................39
Automation.c.............................................................................40, 46
automation.config.........................................................................41
Automation.config.........................................................................47
automation.css.............................................................................46
Automation.css.............................................................................46
Automation.glade...........................................................................16, 46
Automation.glade.............................................................................46
Broadcast directory......................................................................43
CAM_INSET_HGT.............................................................................48
CAM_INSET_HOR.............................................................................48
CAM_INSET_VER.............................................................................48
CAM_INSET_WID.............................................................................48
Cam0 Button..................................................................................25
CAM0_DELAY..................................................................................48
Camera Delay...............................................................................48
Camera initiation.............................................................................9
Camera Inset...................................................................................48
Camera View...............................................................................35
Casting Area Background Radio Buttons..................................26
Casting Modes..............................................................................26
Chrome.........................................................................................15
Chrome Button..............................................................................26
Chromium....................................................................................15, 32, 43
chromium-browser........................................................................44
Cinnamon.......................................................................................45
compileAutomation.script..........................................................45, 49
Compiling the Code.......................................................................45
compression rates...........................................................................11
compressor.....................................................................................15
Console Placement.........................................................................48
CPU load.......................................................................................41
Default View.................................................................................48
locked during streaming.

Linux Mint 18.3 with Mate

Linux Mint 18.3

Desktop Theme

File Search

File Search and Play Button

File search box

File Select / Randomize / Play Buttons

Full Directory Select

Full Directory Select / Play

Full Screen Button

Full Screen Resolution Mode

gcc/g++ compiler and libraries

grep

GTK Configuration

GTK desktop themes

gtkmm

hide all button

image.jpg

Insert Youtube URL / Play Buttons

install.script

Installation

Installation & Compilation

Installation Script

JACK Audio

libcanberra-dev

libcanberra-gtk3-dev

libgdk-3-0

libgdk-3-dev

libgtkmml-3.0-dev

libpulse-dev

libvdpau1

Linux Base

Linux Mint 18.3

locked during streaming.
<table>
<thead>
<tr>
<th>Term</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulseAudio</td>
<td>36, 39p., 50</td>
</tr>
<tr>
<td>pulseaudio-utils</td>
<td>44</td>
</tr>
<tr>
<td>Rand Program Button</td>
<td>23, 41</td>
</tr>
<tr>
<td>Randomize</td>
<td>17</td>
</tr>
<tr>
<td>Real Time Messaging Protocol</td>
<td>10</td>
</tr>
<tr>
<td>Required Software</td>
<td>43</td>
</tr>
<tr>
<td>runAutomation.script</td>
<td>40</td>
</tr>
<tr>
<td>Running the Console</td>
<td>38</td>
</tr>
<tr>
<td>Saver Button</td>
<td>24</td>
</tr>
<tr>
<td>screen saver</td>
<td>41, 43</td>
</tr>
<tr>
<td>Screen Saver Mode</td>
<td>29</td>
</tr>
<tr>
<td>Screensaver</td>
<td>48</td>
</tr>
<tr>
<td>Script Files</td>
<td>49</td>
</tr>
<tr>
<td>Search Box</td>
<td>21</td>
</tr>
<tr>
<td>Search Boxes</td>
<td>28</td>
</tr>
<tr>
<td>Seq Program Button</td>
<td>42</td>
</tr>
<tr>
<td>Seq Program Button</td>
<td>23</td>
</tr>
<tr>
<td>Server URL and Casting Key</td>
<td>39</td>
</tr>
<tr>
<td>Set Up Program.lst</td>
<td>38</td>
</tr>
<tr>
<td>Skype View</td>
<td>31</td>
</tr>
<tr>
<td>skypeforlinux</td>
<td>31</td>
</tr>
<tr>
<td>slide show</td>
<td>41</td>
</tr>
<tr>
<td>Slide Show Mode</td>
<td>30</td>
</tr>
<tr>
<td>Slides Button</td>
<td>24</td>
</tr>
<tr>
<td>Slides640</td>
<td>40</td>
</tr>
<tr>
<td>SOUNDER_MAX</td>
<td>28, 49</td>
</tr>
<tr>
<td>SS_LIB</td>
<td>48</td>
</tr>
<tr>
<td>SS_NAME</td>
<td>48</td>
</tr>
<tr>
<td>Standalone Mode</td>
<td>37</td>
</tr>
<tr>
<td>Starting the Stream</td>
<td>41</td>
</tr>
<tr>
<td>Stop Players</td>
<td>42</td>
</tr>
<tr>
<td>Stop Players Button</td>
<td>23</td>
</tr>
<tr>
<td>stream</td>
<td>40</td>
</tr>
<tr>
<td>Stream Inactive</td>
<td>40</td>
</tr>
<tr>
<td>stream.script</td>
<td>36, 39p., 50, 52</td>
</tr>
<tr>
<td>streaming server host site</td>
<td>39</td>
</tr>
<tr>
<td>t-spi2-core</td>
<td>44</td>
</tr>
<tr>
<td>Video Button</td>
<td>25</td>
</tr>
<tr>
<td>Video Decode and Presentation API for UNIX</td>
<td>8</td>
</tr>
<tr>
<td>VLC</td>
<td>15, 20, 44</td>
</tr>
<tr>
<td>wmctrl</td>
<td>44</td>
</tr>
<tr>
<td>Workspace Switching</td>
<td>11</td>
</tr>
<tr>
<td>XFCE</td>
<td>45</td>
</tr>
<tr>
<td>YouTube Button</td>
<td>19, 26</td>
</tr>
<tr>
<td>YouTube Fill Button</td>
<td>19</td>
</tr>
<tr>
<td>YouTube URL</td>
<td>19</td>
</tr>
<tr>
<td>Youtube URL button</td>
<td>42</td>
</tr>
</tbody>
</table>
YouTube Videos.................................................................................................................................19
YouTube View.........................................................................................................................................32
/dev/video0...........................................................................................................................................35
/usr/lib/xscreensaver............................................................................................................................29
$HOME....................................................................................................................................................39
$HOME/Broadcast/Slides640....................................................................................................................40
$HOME/Broadcast/Slides640/..................................................................................................................30
$HOME/Desktop/Automation....................................................................................................................39
$HOME/Desktop/Broadcast.....................................................................................................................21, 39