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/
Copyright (C) 2018 by Kevin C. O'Kane

This program described herein is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software Foundation;
either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program;
if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA.

March 29, 2018
# Table of Contents

1 Overview..........................................................................................................................5  
  1.1 Requirements............................................................................................................5  
  1.2 Purpose.....................................................................................................................5  
  1.3 Basic Design............................................................................................................6  
  1.4 Performance.............................................................................................................6  
  1.5 Workspace Switching...............................................................................................7  
  1.6 Implementation.........................................................................................................7  

2 The Automation Console.....................................................................................................8  
  2.1 Overview...................................................................................................................8  
  2.2 Basic Layout..............................................................................................................8  
  2.3 Casting Modes..........................................................................................................13  
    2.3.1 Default Cast........................................................................................................13  
    2.3.2 Panel Cast..........................................................................................................14  
    2.3.3 Full Screen Cast..................................................................................................14  
  2.4 Screen Saver Mode....................................................................................................14  
  2.5 Slide Show Mode.......................................................................................................15  
  2.6 Skype View................................................................................................................16  
  2.7 YouTube View..........................................................................................................17  
  2.8 OBS View................................................................................................................17  
    2.8.1 OBS Settings......................................................................................................18  
  2.9 Camera View............................................................................................................19  

3 Encoder..............................................................................................................................20  

4 Audio System....................................................................................................................20  

5 Standalone Mode.............................................................................................................20  

6 Installation........................................................................................................................20  
  6.1 Linux Base................................................................................................................20  
  6.2 Desktop Theme.........................................................................................................21  
  6.3 Installation................................................................................................................21  

7 Running the Console.........................................................................................................22  
  7.1 Setting Up Program.lst............................................................................................22  
  7.2 Server URL and Casting Key.....................................................................................23  
  7.3 Directories................................................................................................................23  
  7.4 Running the Console.................................................................................................23  
  7.5 Installation & Compilation.........................................................................................24  
    7.5.1 Required Software..............................................................................................24  
    7.5.2 Installation Script...............................................................................................25  
    7.5.3 Compiling the Code............................................................................................26  
    7.5.4 Modified Open Source Code............................................................................26  

8 Options.............................................................................................................................26  

9 Script Files........................................................................................................................26  
  9.1 install.script...............................................................................................................26  
  9.2 compileAutomation.script.......................................................................................27  
  9.3 loopback.script.........................................................................................................27
Illustration Index

Figure 1 Automation Console.................................................................9
Figure 2 Automation Console With Video..............................................10
Figure 3 Screen Saver Mode.................................................................15
Figure 4 Slide Show Mode....................................................................16
Figure 5 Skype View............................................................................16
Figure 6 YouTube View........................................................................17
Figure 7 OBS View.............................................................................19
Figure 8 Camera View With Special Guest!.........................................20
1 Overview

1.1 Requirements
This system has been developed and tested on Linux Mint 18.3 with Mate. Since the system makes use of window manager facilities, it may not work correctly on other versions of Linux with other window managers. Generally speaking, however, an Ubuntu derived Linux distribution with a GTK 3+ based window manager may work. If not, the display will not be correctly organized and buttons will overlap on another.

1.2 Purpose
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 builtin `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.

### 1.3 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 and YouTube video controls.

### 1.4 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.

### 1.5 Workspace Switching

If you switch workspaces while casting with the control panel, audio will continue to be streamed although the visible casting area will switch to the new workspace. See below in section 2.3 for details on the size of the areas being cast.

### 1.6 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 may work with other versions of Linux. The implementation depends upon the underlying window manager for many services and these services may or may not be available in other Linux distributions.

---

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

2.1 Overview
The system consists of a casting console that controls the audio and video presented to a casting area of the screen. The casting area of the screen (the red rectangle with the legend 3 in Figure 1 Automation Console) is 640 wide by 350 high and located in approximately the center of your system screen. Visuals dragged into this area will appear in your transmission.

2.2 Basic 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.

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\(^2\)) 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).

\(^2\) 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.
Both the **Mic** and **Mon** meters 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 locations.
3. The casting or canvas area - the rectangle the upper right of the panel - where videos, slide shows, the camera, and the screen saver will display. The area of the rectangle is 640 by 360. This is the basic cast resolution for the present. Any image or video dragged to this area will be transmitted when the stream encoder is active.

If there is no content to display (video, slide show, screen saver), the gray rectangle will normally appear identifying the location of the casting area. You may select another image by clicking one of the radio buttons that appear beneath the casting area.

Any visual content located within the casting area of the screen is presented to the encoder (if active) for transmission.

4. The master system audio control (lower left panel with tabs) showing, in Figure 2, the output level and volume control for the mpv player which is playing the video clip seen in the casting area. 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, the Skype or Chrome controls will appear here. The audio control panel is generated by a modified version of the standard Pulse Audio program pavucontrol which is provided in the distribution.

5. The search box below the microphone volume level (R Mic) is used to search all the files in the Broadcast directory and any sub-directories thereof. A list of possible matches will be displayed. A file selected from this list will appear in the text box with the shadow text music search result. Clicking the adjacent Play button will play this file. The search expression may be a regular expression.

6. YouTube URL Two text entry areas (upper left) where YouTube URLs may be pasted (but not typed). The YouTube video whose URL is placed here will be played in a small Chrome window that will appear in the casting area when the adjacent Play button is clicked. The YouTube button inserts the main YouTube URL into the first YouTube text box. If you click the corresponding Play button, the top (searchable) page of YouTube will appear in a small Chrome window in the casting area.

7. Stream on/off button toggles 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 button invokes 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

---

3 Other resolutions are possible but require recompilation.
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.
so.

8. **File Select** There are three **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. You may navigate from there 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 used to select the file will cause the **File Chooser** to open in the directory from which the selection was made however you may navigate to other directories if you wish.

9. **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 use of a corresponding **File Select** button.

10. **Play** The **Play** buttons cause the named file, URL, or playlist to play.

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

12. **Shuffle** The **Shuffle** button causes the selected playlist to be played in random order. On the other hand, the adjacent **Play** button causes the playlist to be played sequentially.

13. The text box with the shadow text **Search program entries** will search the file names of the entries read from the program.lst\(^5\) file and display a list of files meeting the search criteria. The search criteria may be a regular expression. If you select a file from the list presented, the large program buttons (see below) will reorganize such that the file selected is in the upper left hand program box.

14. To the right of the search box is an area where the time of the currently playing file will appear. The first number is the amount of time in minutes and seconds that have been played and the second is the total time of the file.

15. The three columns of large buttons in the center of the control panel are the program

---

\(^5\) The file program.lst is a text file containing the absolute file system paths to media files, one per line, that populate the twelve large buttons in the center of the console.
buttons. Each button shows the name of a file read from the file \textit{program.lst}. The file names are shown with path information omitted (the \textit{program.lst} file must containing 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 clicked moves into the first position (upper leftmost position).

16. \textbf{Seq Program} The \textbf{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 \textit{program.lst} which may be organized into a program play list.

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

18. \textbf{Clear Program} The \textbf{Clear Program} button halts program play and closes the player.

19. \textbf{Pause} The \textbf{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.

20. \textbf{Origin} The \textbf{Origin} button returns the program buttons to the beginning of the \textit{program.lst} file.

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

22. \textbf{Prog <} This button shifts the program button page one page to the left.

23. \textbf{Prog >} This button advances the program button page one page to the right.

24. \textbf{Skype} The \textbf{Skype} button starts the Skype program (\textit{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.

25. **Show Screen Saver** Displays the screen saver in the casting area. The default screen saver is the *noof* screen saver normally distributed with Linux. If *noof* 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`.

26. **Show Slide Show** 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.

27. **Show Players** Causes any currently running player (Chrome app or MPV) to appear in the casting area.

28. **Camera On** Turns the camera on (or restores it if not already on) and places the video stream into the casting area. 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.

29. **Hide All** Hides all casting area content and raises the casting area frame.

30. **Move Meters** moves the one or both (compilation option) volume meters to the casting area.

31. Below the **Move Meters** bottom are two numbers. The first is the index of the currently playing file in the *program.lst* file and the second is the total number of entries in the file.

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

### 2.3 Casting Modes

#### 2.3.1 Default Cast

In default cast mode, 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 on. The resolution of the cast will be 640 wide by 360 high. When `ffmpeg` is on, you must not move the console window. Note: when you engage `ffmpeg`, it locks onto the gray casting area.
coordinates. You may move the console prior to engaging ffmpeg but once engaged, the casting coordinates are fixed until you end ffmpeg.

2.3.2 Panel Cast
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.

2.3.3 Full Screen Cast
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 a modest 1100 k bits/second.

2.4 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 preformatted slideshow.

The screensaver mode used here consists of a display of animated patterns taken from the xscreensaver package which is part of many Linux distros. 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.

---

6 See the symbols SS_LIB and SS_NAME in Automation.c
2.5 Slide Show Mode

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

You should run and set Phototonics’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, and initial image is required. The default is:

image.jpg

So, you should ensure that the directory has an initial image named image.jpg.
2.6 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.
2.7 YouTube View

When the Play button adjacent to a YouTube URL is clicked, a small Chrome 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 Chrome app window from which you can search and play videos.

![Figure 6 YouTube View](image)

2.8 OBS View

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 7 (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
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 bandwidth to produce a cast of similar quality to the builtin ffmpeg encoder.

### 2.8.1 OBS Settings

The OBS configuration shown in Figure 7 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 your casting key in the OBS settings.

### 2.9 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 camera 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:

```
   guvcview --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 8 Camera View With Special Guest!](image)

3 Encoder

4 Audio System

5 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.

6 Installation

6.1 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 should work on other distros and window managers but it has not been tested as of this time.

### 6.2 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.

### 6.3 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 program (from synaptic) mpv

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
7. bashrc

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

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

11. See Section 7.3 for information on what directories are required and their contents.

7 Running the Console

7.1 Setting 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

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.
7.2 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 whatever other service you intend to use that supports the rtmp protocol. This information should be available from the streaming server host site.

7.3 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.

7.4 Running the Console
You initiate the system with the command:\n
```
./runAutomation.script
```

8 Note: the ./ is not required if you included the current directory in PATH as shown above.
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 built-in 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. 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 Stream on/off toggle button 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.

### 7.5 Installation & Compilation

#### 7.5.1 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 (xscreensaver) is installed. The system uses the noof 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. pqiv

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

### 7.5.2 Installation Script

The process of installing missing items from the 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 6.3 on page 21 above.
7.5.3 Compiling the Code

Once you have installed the required software, you should now be able to compile the system with the command:

```
./compileAutomation.script
```

7.5.4 Modified Open Source Code

Included with the distribution are compiled versions of `pavumeter`, `pavucontrol` and `guvcview`. 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 `guvcview` (for `guvcview`). Copy the executables to the main Automation directory. The `configure` procedure for these may identify additional software requirements.

For `pavumeter` and `pavucontrol`, follow the procedure outlined in section 6.3 on page 21 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 Options

For the most part, options are set in `Automation.c` as variables and defined symbols.

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

9 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
```

9.1 `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.

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

9.3 loopback.script
This file sets up the PulseAudio loopback that captures audio for streaming. If you are using OBS for streaming, it is not necessary to execute this file. The audio can be configured directly in OBS.

If you are using the built-in streaming facility, you need to execute this file. The main execution script, runAutomation.script, automatically executes this file.

The script sets up an 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 is used as the input to the streaming encoder software. Otherwise, the encoder would only receive audio from the microphone or the players but not both.

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 can become confused 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 Hardware tab. The devices will remain disabled until you turn them back on again.

9.4 stream.script
The file stream.script contains the encoder commands. It is automatically executed by 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).

9.5 runAutomation.script
The file runAutomation.script executes loopback.script and runs the automation system. The automation system will invoke stream.script only if you click the stream on/off button.
10 FAQ

10.1 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 \texttt{mpv} to silent, it will remain at silent until you change it - even if you reboot!

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