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 22, 2018
# Table of Contents

1 Overview.................................................................................................................................4  
  1.1 Purpose.................................................................................................................................4  
  1.2 Basic Design..........................................................................................................................4  
  1.3 Implementation......................................................................................................................5  
  1.4 Performance..........................................................................................................................5  
2 The Automation Console..........................................................................................................6  
  2.1 Overview.................................................................................................................................6  
  2.2 Basic Layout............................................................................................................................6  
  2.3 Screen Saver Mode..................................................................................................................10  
  2.4 Slide Show Mode...................................................................................................................10  
  2.5 Casting Area Frame...............................................................................................................12  
  2.6 Panel View..............................................................................................................................12  
  2.7 Skype View............................................................................................................................13  
  2.8 YouTube View......................................................................................................................13  
  2.9 Browser View.......................................................................................................................14  
  2.10 OBS View............................................................................................................................15  
  2.11 Camera View.......................................................................................................................16  
  2.12 Standalone Mode..................................................................................................................16  
3 Installation...............................................................................................................................16  
  3.1 Linux Base.............................................................................................................................16  
    3.1.1 Setting the Desktop theme...............................................................................................16  
  3.2 Quick Binary Distro Install....................................................................................................17  
  3.3 Installation & Compilation.....................................................................................................18  
    3.3.1 Required Software............................................................................................................18  
    3.3.2 Compiling the Code..........................................................................................................19  
4 Options......................................................................................................................................20  
5 Script Files.................................................................................................................................20  
  5.1 install.script...........................................................................................................................20  
  5.2 compileAutomation.script.....................................................................................................20  
  5.3 Running the System..............................................................................................................20  
    5.3.1 loopback.script................................................................................................................21  
    5.3.2 stream.script...................................................................................................................21  
    5.3.3 runAutomation.script......................................................................................................21  
6 FAQ...........................................................................................................................................21  
  6.1 No Sound...............................................................................................................................21  
  6.2 No Sound, Yet Again.............................................................................................................21
1 Overview

1.1 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, with a 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.

1.2 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. Slideshow, screen saver, camera and YouTube video controls.

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. An 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 soon be available).

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 and full motion, 24 fps video, at 640x350 can be encoded and transmitted as bit rates less than 850K with no graphical video card requirements. The system is suitable for laptops and portable applications.

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.

### 1.3 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 distros.

### 1.4 Performance

On an AMD fx 6100 processor overclocked at 4 gHz, 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: varies but less 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.
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 Automation console with a video playing and a camera running.

The active areas of the console are as follows:

1. The internal system volume meters L Mon and R Mon (pavumeter) showing the sound levels being presented to the encoder.

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

3. The casting area - the rectangle to the right of the control panel where a video clips is displayed. Note: if there is not content to display, a red rectangle will appear showing the location of the casting area. The red rectangle may be restored by clicking the Hide All button. The contents of the casting area of the screen will be presented to the encoder (if active) for transmission.

4. The video camera (or cameras) (/dev/video0) appears in a window above the casting area. The cameras may be resized and dragged into the casting area as needed. When cameras are in the casting area, their contents are presented to the encoder, if active, for transmission. The contents of the cameras are not included in the transmission if they appear outside the casting area.

5. The master system audio control (lower left panel with tabs) showing, in this case, 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, the Skype control will appear here. The audio control panel is generated by a modified version of the program pavucontrol provided in the distro.

6. In the upper right corner are two volume meter displays. The top display is the volume being presented to the encoder and the second is the level of the microphone.
7. The search box below the microphone volume level searches all the files in the `Broadcast` directory. 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.

8. **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.

9. **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 so.

10. **Cam0 / Cam1** Two buttons that toggle on/off the cameras if available. Cameras are assumed to be named `/dev/video0` and `/dev/video1`. These buttons invoke a modified version of `guvcview` (provided in the distro, see below) and the camera stream(s) are...
initially placed below the control panel in small windows. Either or both of the camera windows may be dragged to the casting area and resized. Note: right clicking on the small icon on the left hand side of the camera window task bar will allow you to set a camera to be always on top so that the cameras will be above any new casting area content.

11. **File Select** There are three *File Select* buttons. Each opens a standard *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 (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.

12. **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, from which a file will be randomly chosen, has been identified by use of a corresponding *File Select* button.

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

14. **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.

15. **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.

16. The text box with the shadow text *Search program entries* will search the file names of the entries read from the *program.lst* 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 next topic) will reorganize such that the file selected is in the upper left hand program box.

17. The 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.

18. The three columns of large buttons in the center of the control panel are the program
buttons. In each button the name of a file read from the file program.lst is shown. The file names are shown which path information omitted (the program.lst file must containing path information - see below).

Each file name appears with (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 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).

19. **Seq Program** 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.

20. **Rand Program** 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.

21. **Clear Program** The **Clear Program** button halts program play and closes the player.

22. **Pause** 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.

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

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

25. **Prog <** This button shifts the program button page one page to the left.

26. **Prog >** This button advances the program button page one page to the right.

27. **Skype** The **Skype** button starts the Skype program (skypeforlinux) and places the Skype window in the upper right corner of the screen.
28. **Home** Clears the casting area, moves the casting control panel to its original position, and raises the casting area red frame.

29. **Panel View** moves a section of the panel into the casting area thus displaying the currently playing media file, the time remaining, the stream bit rate and the volume meters to the encoder.

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

31. **Show Slide Show** Initiates the slide show in the casting area.

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

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

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

35. Below the **Move Meters** button 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.

### 2.3 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 presented a compiled option but selections buttons should be available soon.

### 2.4 Slide Show Mode

The slide show is a display of images that part of the cast. The image presenter is *Phototonics*. You should set Phototonic’s preferences before you use it. Ideally, slide show should be set to *random*, time to 10 seconds per slide, and not to display file name in the slide.
Figure 2 Screen Saver Mode

Figure 3 Slide Show Mode
2.5 Casting Area Frame

The casting area frame (in red) shows the area of the screen that is cast. Any visual, still or animated, dragged into this are will be cast but only the portion within the frame.

![Casting Area Frame](image)

Figure 4 Casting Area Frame

2.6 Panel View

In Panel View, the control panel moves to the casting frame, the volume meters move to the lower left and the volume control panel moves to the upper left of the monitor screen. The area of the panel within the casting frame ranges from the stream bit rate down through the volume meters. The currently playing program selection is visible.
2.7 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.

![Skype View Image]

2.8 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.

![YouTube View Image]
2.9 Browser View

In the example, a Chromium display of a NOAA forecast has been dragged within the casting rectangle. This content will be transmitted. The area outside the red rectangle is not seen by viewers.
2.10 OBS View

The system can be used as a set of source windows for OBS where OBS can be used for recording or encoding. In this view, the main panel, the microphone and system volume levels, the audio control panel and the player (mpv in this example) are each separate windows that may or may not be captured by OBS.
2.11 Camera View

2.12 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.

3 Installation

3.1 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.

3.1.1 Setting the 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.

3.2 Quick Binary Distro Install

1. The distro comes with binary executables which are compatible with Linux Mint Mate 18.3 and probably related Ubuntu based systems. They should work 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 distro in you Desktop directory

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)

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 to .bashrc

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

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

11. 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:
The above is an absolute file address in my system. You can get a list of absolute addresses on your system by:

1. Open a **caja** file explorer window to show the file names in 'compact' format (edit preferences).
2. Type `^a` followed by `^c` (means **all**, and **copy**)
3. Open the text editor (**xed**) from Applications | Accessories
4. Type `^v` (paste). The edit window will now contain the file names. 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.

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

1. To cast: modify the SERVER and KEY settings in **stream.script** to point to **ivlog** or **vaughnlive** or whatever other service uses **rtmp** protocol.
2. You initiate casting by clicking the Stream on/off toggle.
3. Terminate the window by clicking the X box, upper right corner.

You initiate the system with the command:

```
./runAutomation.script
```

As the system comes up, 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. Note: on some slower systems, the delays may not be sufficient. You will also see messages. Some will be minor error messages and this is normally not a problem. If it crashes, they were problems ....

### 3.3 Installation & Compilation

#### 3.3.1 Required Software

The system uses a number of freely available software packages, many of which may already be installed and no further action is required. We assume that the standard Linux Mint screen saver is installed as the system will attempt to use the **noof** saver (this is a setting in **Automation.c**)

The script file **install.script** run as:
sudo ./install.script

will download and install the needed software. This script will generate a lot of messages and may take some time to run. You must have a functional Internet connection to use it.

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

They are available in the Synaptic package manager.

3.3.2 Compiling the Code

Once you have installed these, you should be able to compile the system with the command:
To recompile the modified **pavumeter** provided in the distro (probably not necessary), descend into that directory and then, as root:

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

The executable will be in the `src` sub-directory. Copy it to the main Automation directory.

To recompile the modified **pavucontrol** provided with the distro, descend into the `pavucontrol` directory and repeat the steps shown for **pavumeter**.

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

## 4 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`

## 5 Script Files

The distro comes with several BASH script files. These all have a `.script` file extension. They are used to install, compile and run the programs.

### 5.1 `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.

### 5.2 `compileAutomation.script`

This script compiles the system code.

### 5.3 Running the System

There are three script files used to run all or part of the system. These all need to be user executable (should be the default). To make a file executable, type:

```
chmod u+x filename
```
5.3.1 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.

5.3.2 stream.script
The file stream.script is the encoder. It is automatically executed by the runAutomation.script file.

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

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

5.3.3 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.

6 FAQ

6.1 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 to silent, it will remain at silent until you change it - even if you reboot!

6.2 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.