Manual for Pi Presents

© Ken Thompson, 2013

for Version 1.2.1 beta 1, 9 April 2013

1 Introduction

Pi Presents is a presentation application intended for Museums and Visitor Centres. I am involved with a couple of charity organisations that are museums or have visitor centres and have wanted a cost effective way to provide audio interpretation and slide/videoshow displays. Until the Raspberry Pi arrived buying or constructing even sound players was expensive. The Raspberry Pi with its combination of Linux, GPIO and a powerful GPU is ideal for black box multi-media applications; all it needs is a program to harness its power in a way that could be used by non-programmers.

Pi Presents offers quite complex features with simple to use editing facilities to configure it to your exact display requirements. The features include:

- Animation or interpretation of exhibits by triggering a sound, video, or slideshow from a button, keyboard or GPIO inputs.

- While playing media GPIO outputs can be used to control external devices such as lights, moving machinery, animatronics etc.

- A repeating media show for a visitor centre. Images, videos, audio tracks, and messages can be displayed. Repeats can be at intervals or at specified times of day.

- Allow media shows to be interrupted by the visitor and a menu of videos to be presented.

- Giving talks where progress through slides is manually controlled by buttons or keyboard. The presentation may include images, text, audio tracks and videos.

- Using Liveshow dynamically upload image, video or audio tracks over the network for display in a repeating show.

- Run multiple shows simultaneously so that more than one exhibit can be controlled from one Pi or complex interactive exhibits can be controlled.

- Foreign language support.

There are potentially many applications of Pi Presents and your input on real world experiences would be invaluable to me, both minor tweaks to the existing functionality and major improvements.

Once set up, use of Pi Presents is easy to use and does not require a network:

- Shows can be prepared using a simple to use editor on a Windows PC or a Linux box.

- Will operate on a Model A Raspberry Pi (but see hardware requirements).

- No need to modify the Pi's SD card after initial installation. All media and configuration options can be kept on a removable USB stick. Installing a new application is as simple as inserting the USB stick.

- Control can be by buttons, PIR, or any source of digital inputs. It is straightforward to make your own controls in a box containing the Pi, or perhaps a wired remote. Alternatively a keyboard can be used.

- Video track is selectable between HDMI or composite using the normal Pi setup procedure.

- Audio track output is selectable between HDMI and analogue and can be output from left, right or stereo speakers on a per track basis. A number of tracks can be played simultaneously.

Pi Presents, out of the box, runs as a desktop application on the Raspberry Pi using the keyboard for control. However with a little bit of Linux magic it can be made to run as a black box application with control from GPIO; full instructions are given below. Black box features include:

- Disabling screen blanking

- Full screen operation without window decorations
- Black screen behind videos and images
- Completely headless operation without a keyboard, mouse or buttons (except perhaps a shutdown button if you are not brave)
- Optional operation with pushbuttons or PIR through GPIO. Input and output pins are fully configurable.
- Automatic start up when power is applied to the Pi
- All media can be on a USB stick or on the SD card
- Safe shutdown without using keyboard or mouse

While Pi Presents is easy to set up for the majority of tasks using the editor application; it also very flexible allowing complex combinations of media shows to be produced and triggered.

1.1 Acknowledgements

Johannes Baiter (jbaiter) for pyomxplayer based on Noah's pexpect

Numerous people on the Raspberry Pi forum and websites, particularly StackOverflow, who have unwittingly given solutions to many technical problems and taught me Python.

ActiveState for a free Python systems for Windows

The Raspberry Pi Foundation for the first affordable machine that plays video, audio and has GPIO.

2 Installation

The installation instructions are in the README.md file.

3 Try Some Examples

Download the examples from the pipresents-examples github repository as described in the README.md file.

From a terminal window opened in the pipresents directory type:

```
python pipresents.py –p pp_mediashow
```

You will see a continuous looping show.

The pipresents-examples repository has examples of how to use Pi Presents:

- pp_mediashow
  The profile you have just run. The show will start immediately, progress automatically, and repeat after 70 seconds. Use the Up and Down cursor keys to skip.

- pp_menu
  Use the Up and Down cursor keys to traverse the menu, Return to Start a track, and Escape to terminate it.

- pp_exhibit
  This demonstrates how to trigger a mediashow from a PIR or button. When started, the screen will be blank. Pressing the Play button or triggering the PIR will start a one shot mediashow and then wait for another trigger. The example uses the Return key instead of the PIR to see the effect.

- pp_interactive
  This combines a mediashow and a menu. The mediashow is run continuously but every track has a hint showing that a menu can be triggered by the Return key. Pressing Escape from the menu returns to the mediashow.

- pp_presentation
  The mediashow is configured as a presentation. Use the Return key to start the presentation and the Up and Down cursor keys to traverse through it.

- pp_liveshow

  When started you will see an introduction followed by the blank screen of the LiveShow. While Pi Presents is running place some video files, audio files, or images into the directory /pp_home/pp_live_tracks.

- pp_audio

  This application demonstrates the capabilities of the new audioplayer, you will need speakers connected to hdmi and analogue ports to appreciate it fully.

- pp_concurrent

  This application demonstrates concurrent show playing capabilities. The mediashow can be controlled from buttons or keyboard while audio tracks are played continually.

- pp_timeofday

  Triggers a mediashow and a liveshow at a specified times of day. To make this demonstration work I have replaced the times of day with delays (+xx). You can replace these test delays with real local times e.g. 10:30.

All examples use a selection of media tracks which are in /home/pi/pp_home/media. The profiles are in /home/pi/pp_home/pp_profiles. The pp_showlist.json file specifies the look and feel of the show; other files specify the media to use. The files can be viewed in a text editor.

The keyboard commands are:

- Up or Down Cursor- move through a menu.
- Up or Down Cursor - Move through a presentation type show. Can also be used to skip to the next track or previous in an automatic mediashow. The movement circles around at the start and end of a show.
- Return – Start a presentation or play the selected menu entry.
- Escape - In a track, Stop and return to the show. In a show, back to the calling show.
- In an image, video, or audio track p or spacebar- pause/resume.
- CTRL-BREAK always exits Pi Presents

4 The Pi Presents Profile Editor

Profiles can be created and edited using the pp_editor.py

```
python pp_editor.py
```

4.1 The Displays

Select one of the example profiles using the Profile>Open menu. Gotcha: Remember to open the selected profile directory before pressing OK.

- The top left panel displays the shows in the profile
- The bottom left panel shows the medialists in the profile, click on one of the entries to select it.
- The right-hand panel shows the tracks in the selected medialist.

The selected entries are shown in red. Click the 'Edit Show' button adjacent to the left-hand panel to edit the selected show and the 'Edit' button adjacent to the right-hand panel to edit the selected track.

Edits are saved to disc when OK is pressed, use Cancel if you want to duck out.

You can run pipresents.py and pp_editor.py concurrently from two terminal windows so the effect of any edits can quickly be seen. You will need to restart pipresents.py to see the updates. If you are going to edit the examples make a copy of them first by using the File Manager to access /home/pi/pp_home/pp_profiles and to copy a directory.

Section 10 of this manual contains a complete definition of all of the types of shows and tracks.

4.2 Profile Menu Operations

Profile>Open - displays a directory viewer to select a profile for editing. The profiles displayed are those in the home directory defined in the Options>Edit menu.

Profile>Validate - Validate the Profile. Font and Colour fields are currently not validated and if you edit the .json files with a text editor you are on your own!

Profile>New from Template - lists templates for all the example shows. The templates have example tracks so they will run un-edited.

4.3 Shows

Shows can be of three types - menu, mediashow, and liveshow. Mediashows play a sequence of tracks; menus allow tracks to be selected by the user. Liveshows allow tracks to be dynamically added and displayed in sequence.

All shows have the following fields:

- title - Text displayed in the editor and on a menu

- show_ref - A label which allows other shows to reference this show. Can be any alpha-numeric string without spaces.

- medialist - this is the name of a file which appears in the medialist panel. It must have the extension .json. Every show must have a medialist to define the tracks in the show.

Shows have fields which control the sequencing and look of the show. They also have a number of fields which provide default values for all the tracks in the show e.g. duration, transition, omx-audio; tracks will use the values in the show if their corresponding fields are left blank.

The tracks in a show can themselves be shows, so a mediashow could include segments which are themselves mediashows, a menu of mediashows could be produced, or you could have a menu of menus. The permutations are infinite! These are called 'sub-shows'; sub-shows are tracks so appear in the medialist.

4.3.1 Mediashows

Think of a mediashow as a slideshow that can play tracks (slides) of different types - videos, audio tracks, images, and even animation control sequences.

Mediashows have a number of fields to define the shape for the show, e.g. trigger, progress, repeat and repeat-interval.

A menu can be associated with a mediashow such that the menu is accessible from any track in the show. These are termed 'child shows'. The 'has-child' field tells Pi Presents that a menu is associated with the show; the menu itself is referenced in the medialist for the show. Associated with the use of a 'child show' is 'hint-text' that is displayed only when 'has-child' is yes.

Mediashows are at the heart of Pi Presents, there is more information on their control in Section 5.1

4.3.2 Menus

A menu has a number of fields to define the look and feel of the menu e.g entry-x, entry-y, entry-spacing, entry-font etc.

A menu needs some instructions for the user this is included in the hint-text field.

If a menu is not used for a period of time it might be useful to return to the calling mediashow, the timeout field, if non-zero, provides the function.

A menu can have a background image, The 'has-background' field enables this. The reference to the menu is in the medialist associated with the show.

4.3.3 Liveshows

Liveshows repeatedly display tracks in alphabetical sequence of filenames.

A menu can be associated with a liveshow such that the menu is accessible from any track in the show. These are termed 'child shows'. The 'has-child' field tells Pi Presents that a menu is associated with the show; the menu itself is referenced in the medialist for the show. Associated with the use of a 'child show' is 'hint-text', this is displayed only when 'has-child' is yes.

A medialist associated with the liveshow is used to specify the child-show and anonymous tracks in it will be ignored. A medialist must be associated with a Liveshow even if there is no child-show.

4.3.4 Start Shows

The Start show specifies the shows to be run in the 'start-show' field. All shows specified in this field will be run when Pi Presents starts. Each show will then wait for its trigger (which may be Start, so no waiting) and run concurrently with other shows.

There are limits to combinations of resources that can be used with concurrent shows. These are described in Section 5.2

4.3.5 Show Menu Operations

Show>Add - Add a new show. The show type of an existing show cannot be edited.

Show> Edit - Duplicates the Edit Show button

Show>Remove - Deletes the show from the profile, no going back!

4.4 Medialist Operations

Medialist>Add - Add a new medialist. The .json extension is added if not included.

Medialist>Remove - Deletes the medialist from the profile, no going back!

4.5 Tracks

Medialists are similar to playlists in a media-player in that they define the tracks to be played either in sequence or as part of a menu.

4.5.1 Anonymous and Labelled tracks

Anonymous tracks have a blank 'track-ref' field. Anonymous tracks are included in a menu or mediashow in the order that they are displayed in the editor.

Tracks with a label are not included in a menu or mediashow. They are used for special purposes and have special labels.

4.5.2 Media Tracks

Media tracks play things. They reference the file containing the media and provide per-track definition of how the track is played.

Media track types are:

- Video
A track played by omxplayer. Pi Presents should play any track that omxplayer can play (but see hardware requirements). Video tracks can be paused with the Pause GPIO button or the 'space' or 'p' keys. Animation facilities are available.

- Audio
A track played by Mplayer. Pi Presents should play any audio only track that Mplayer can play. Audio tracks can be paused with the Pause GPIO button or the 'space' or 'p' keys. Audio tracks can be overlaid with text; track-text is overlaid on a per track basis, show-text per show; they can also have images as a background. Sound can be directed to HDMI or analogue and to either of the analogue speakers. Animation facilities are available.

- Image
Image tracks are rendered by the Python Imaging Library. (See hardware requirements for recommended image size). Image tracks can be paused with the Pause GPIO button or the 'space' or 'p' keys. Image tracks can be overlaid with text; track-text is overlaid on a per track basis, show-text per show. Animation facilities are available.

- Message
Message tracks display text against a coloured background or optional image. They do not need a file to be specified as the text is contained in the track entry in the profile.

Each type of track has fields describing how to display the track, some of these override the corresponding fields in the calling show.

4.5.2.1 Track File Names

I wanted to produce a system where profiles and their media can be moved between different drives without breaking the references between shows and their media tracks. This is necessary if a profile is prepared on a Pi on a SD card and moved to a USB Stick for operational use; also if the profiles are to be prepared on a Windows machine.

To achieve this portability media tracks should be stored under the /pp_home directory, in sub-directories if desired. If tracks are copied to this location before the profile is prepared then the Profile Editor will automatically compute the appropriate track reference:

- If the file is below the home directory (…../pp_home), as defined in the Options>Edit menu then the file reference will look like '+/track_to_play.mp4'

- Otherwise the reference will look like '/home/pi/mymedia/track_to_play.mp4'; an absolute reference.

Absolute references do have their uses, for example in specifying internet url's e.g. http://www.mysite.com/ track_to_play.mp4

4.5.3 Show Tracks

The tracks in a show can themselves be shows, so a mediashow could include segments which are themselves mediashows, a menu of mediashows could be produced, or you could have a menu of menus. This is called a 'sub-show'.

Show tracks allow sub-shows to be specified. The sub-show field specifies the label (show-ref) of the show to play.

4.5.4 Labelled Tracks

Labelled tracks are used for special purposes. Two preset labels are currently defined:

- **Child Show Tracks**
  These are show type tracks with the show-ref field value of 'pp-child-show'. They are needed in the medialist of a mediashow if 'has-child' is yes. The child-show field is usually refers to a menu but a mediashow could be used.

- **Menu Background Tracks**
  These are menu-background type tracks with the show-ref field value of 'pp-menu-background'. They are needed in the medialist of a menu if 'has-menu-background' is yes.

4.5.5 Track Menus

Track>Add from File
Adds one or more track entries containing media files to the end of the medialist. These may be images, videos or audio tracks. The editor automatically calculates the file location and track type. If either of these is unacceptable then use Track>New to add a blank track of the required type.

The list of extensions that are used to select track type is in the first few lines of the pp_definitions.py source. Please raise a bug report if these are inadequate.

Track>Add from Dir
Similar to Add from File, except that all eligible tracks in the directory are added to the medialist.

The list of eligible extensions that is used to select tracks and their types is in the first few lines of the pp_definitions.py source. Please raise a bug report if these are inadequate.

Track>New
Adds a blank track of the selected type to the end of the medialist.

Track>Edit
Duplicates the Edit Button

Track>Remove
Deletes the track from the medialist, no going back!

4.6 Editor Options

Access from the Options>Edit menu of pp_editor.py

Pi Presents Home Directory
This is an important option which must be set to the directory in which any profile directories and relative media is to be assembled. Out of the box it is set to /home/pi/pp_home. If you choose an alternative setting for this then ensure that the directory /pp_profiles is created inside the chosen /pp_home.

Initial Media Directory
This is just a helper setting to define where 'Add Track' and 'Add from Dir' starts their browsing.

4.7 Editor Command Line Options

-d --debug	Run time errors give alerts and are reported to the terminal window if open. Turning on debugging will in addition provide a trace of the operation of Pi Presents in the terminal window.

	Errors and the trace are also recorded in the /pipresents/pp_log.log file.

4.8 Using USB Stick for Profiles

TBD,

4.9 Using the Editor on a Windows PC

Use on a PC requires a Python system to be installed. I use the free Activestate Python http://www.activestate.com/activepython. There is a good installation guide here http://www.richarddooling.com/index.php/2006/03/14/python-on-xp-7-minutes-to-hello-world/.

To install the editor on a Windows PC download the Zip File from https://github.com/KenT2/pipresents (just click on the Zip button). Extract the contents of the folder pipresents-master to a folder called pipresents.

Either download pipresents_examples and extract the folder to your top level folder (c:\users\ken on my Windows 7 machine) or create a 'pp_home' folder here with a folder 'pp_profiles' inside.

Open pp_editor.py in the Idle editor and run it. Alternatively you can double click on pp_editor.py or create a desktop shortcut to pp_editor.py; you may need associate '.py' files with python.exe.

Use of the editor is the same as on the Pi, it is obviously important that all track references are to media files under pp_home. To run profiles created on the PC just copy pp_home and the relevant profiles and media to the top level of a USB stick and run Pi Presents with the -o option set to /media/STICKNAME.

5 Controlling Shows

With this version the ability to run two or more shows concurrently has been introduced. This has many uses which include:

- Providing a background audio track to a slideshow.
  Use two repeating mediashows, one with the slideshow and the other with the audio tracks. The latter will need the controls disabled if there is and user interaction with the former.

- Running shows at specific times of day

Use two or more liveshows or mediashows each triggered at time of day such that they do not overlap. The liveshow and mediashow can also be stopped at a specific time of day or after a period of time.

- Being really thrifty and doing two completely different tasks at once, perhaps a slideshow in the Reception and a dummy talking in a museum exhibit
- Complex animation where a triggers from GPIO inputs each trigger a mediashow which outputs sound and show animation controls.
- To make shows concurrent more than one 'start-show' should be specified in the Start Show

All shows specified in the 'start-show' field will be run when Pi Presents starts. Each show will run concurrently with other shows. Immediately after starting all shows consider conditions for continuing:

- Menus will always proceed immediately.
- Mediashows with start-trigger = start will run immediately. Mediashows with start-trigger = button, time or GPIO will proceed when the trigger condition is met. Mediashow triggering is explained in more detail in the next section.
- Liveshows with start-trigger = start will run immediately. Liveshows with trigger = time will proceed when the trigger condition is met. Liveshow triggering is explained in more detail in the next section.

In many cases only a subset of the concurrent shows will need to be controlled by keys or buttons so shows have the 'disable-controls' field which:

- If controls are not disabled the show and any sub/child shows will respond to buttons or key presses in addition to other GPIO inputs and time of day triggers.
- If controls are disabled the show will not interact with buttons or keys, it must either proceed immediately or be triggered by time of day or GPIO.

In theory triggers can be applied to sub shows and child shows, I have not tested these extensively. Controls can be enabled in a top level show and then disabled in a child show or sub show; however since controls are passed down from shows to sub/child shows and then to players they will be disabled for all entities under the disabled show.

5.1 Triggering Mediashows and Liveshows

Mediashows have a number of show control facilities; liveshows have a subset of these and references to mediashow in the paragraphs below include liveshow where appropriate. Most of the facilities are designed for use when the show is played without user interaction; however if controls are not disabled they can also be used (with care!)

5.1.1 Start Trigger

Immediately after a mediashow starts it uses the trigger field to decide what to do next, also after each track a trigger can be employed to move to the next track:

The 'start-trigger' can take the following values:

- start
  The mediashow continues to the first track.

- GPIO
  The mediashow waits for a GPIO input other than a button, to trigger running of the first track of the show. The symbolic name of the GPIO input must be included in the 'trigger-input' field. Symbolic names are assigned to pins in the gpio.cfg file (see Section 9.1). Out of the box P1-11 is configured as an input and assigned to the symbolic name PIR to provide continuity with the previous version of Pi Presents.

  GPIO inputs are sent to all shows so if more than one show has the same trigger all will proceed.

- time/time-quiet

  To be able to use time of day either keep the Pi connected to the internet, invest in a hardware real time clock add-on.

  The mediashow waits for a time of day before proceeding. A list of 'times of day' must be included in the 'trigger-input' field.

  Each time of day can specify hour, minute and optionally second as a 24 hour clock and should be separated by a space. Entries may be in any order. e.g.

  ```
  21:2 21:03 9:30 9:30:30 0:0 23:59:59
  ```

  Pi Presents will proceed with the show when the local time reaches one of the times specified in the list. If a show is being played when a time in the list is reached the show will not be interrupted or the need for a show to proceed remembered.

  If there are no shows left in the list for today then the first show in the list will be scheduled and started tomorrow.

  If the option is 'time' and not 'time-quiet' an admin message is displayed while waiting for the show to commence. It is possible to show the time of the next show in the admin messages, see Section 9.2. There is an alternative message for shows starting tomorrow.

Pi Presents may not cope with changes to daylight saving time or any other non-linear changes in local time.

There is an alternative way to specify the time of day. If +10 + 40 is entered instead of, say, 09:30 then the show will start in 10 seconds time then in 40 seconds time from when the mediashow was started. This is aimed at testing only.

- button
  The mediashow waits for the Play button or Return key to be pressed. This is probably only of great use when the show progress is 'manual' and will be operative only if controls are enabled. Its aim is to allow a presenter to pause a presentation before the first track.

5.1.2 End Trigger

Some mediashows that repeat or have whose 'sequence' is 'shuffle' and all liveshows will not end naturally; they need their end to be forced.

This can be useful in some circumstances, for example:

- Running a liveshow throughout the day and then a 'we are closing' mediashow at a particular time of day. Run the two shows concurrently and terminate the liveshow with an end-trigger and use the start-trigger to start the oneshot mediashow.

- Terminating a repeating mediashow or a mediashow that has 'sequence'='shuffle'. The latter cannot be naturally terminated even if it is a one-shot show because, where is the end of a random sequence? In this case the 'duration' option might be preferred over the time of day.

The 'end-trigger' can take the following values:

- none
  The end-trigger is not operative.

- time
  To be able to use time of day either keep the Pi connected to the internet or invest in a hardware real time clock add-on.

  The mediashow waits for a time of day before ending. A list of times of day must be included in the 'trigger-end-times' field.

  Each time of day can specify hour, minute and optionally second as a 24 hour clock and should be separated by a space. Entries may be in any order. e.g.

  21:2 21:03 9:30 9:30:30 0:0 23:59:59

Pi Presents will terminate the show when the local time reaches one of the times specified in the list.

The system should allow shows to proceed over midnight so a show starting at 23:55 to be terminated as 00:05 the next day.

Pi Presents may not cope with changes to daylight saving time or any other non-linear changes in local time.

There is an alternative way to specify the time of day. If +20 + 50 is entered instead of, say, 09:30 then the show will end in 20 seconds time then in 50 seconds time from when the mediashow was started. This is aimed at testing only.

- duration
  The mediashow is terminated after a specified period of time. The time period should be entered in the 'trigger-end-time' field as hour, minute, second.

  The format is h:m:s hour and minute are optional. e.g

  5 - 5 seconds
  6:0 - 6 minutes
  10:0:0 - 10 hours

5.1.3 Next Trigger

The 'trigger-next' field allows individual tracks in a mediashow to be triggered by GPIO. For this to work 'progress' should be set to 'manual':

- none
  The trigger is not operative.

- GPIO
  The show moves to the next track when it is triggered by the input pin of the GPIO having the symbolic name specified in 'next-input'.

5.2 Limitations on Concurrency

Concurrent shows tax the processing power of the ARM processor in the Raspberry Pi and the concurrency built into Linux, MPlayer and OMXPlayer:

- Two shows cannot in general use the screen at the same time.

- If audio tracks to be played with MPlayer might overlap then they should all use the same output either HDMI or Analogue. If not, switching will occur mid-track and I find the system crashes. If you want to play two audio tracks that might overlap you can use Videoplayer to play one of them or just direct each of the tracks to left and right outputs.

- HD videos may stutter while an audio track is being played.

- Animation outputs are delayed if they are coincident with the start of an audio track or image. The output event will not be missed but will be delayed.

6 Remote Operation

Pi Presents was not intended for remote operation however by popular demand I have included a facility where a 'Liveshow' can play tracks dynamically ftp'ed from another computer.

The New> Liveshow template is a working Liveshow. The tracks to be played should be placed in /pp_home/pp_live_tracks.

Tracks can be ftp'ed into the Pi using Filezilla or some such. Using the -l command line option it is possible to have a second location containing live tracks, one in the data home and another in another directory specified by the -l option. The tracks could be on a remote fileserver (I have not tried this, Samba required?).

You may need to set your Pi to use static IP and take network security measures if you use this, as a minimum change the password for user Pi.

7 Animation Control

Most types of track have facilities to control animation. Using commands defined in profiles any GPIO output can be turned on or off synchronised with the start or end of tracks, with optional delay.

An example of animation commands is shown below.

animate-begin	out1 on 1 out1 off 2 out2 on 1 out3 on 6
animate-clear	no
animate-end	out2 off

A command has three fields separated by spaces and terminated with a newline:

- Symbolic Name - The name of the output as defined in gpio.cfg (Section 9.1). There can be different gpio.cfg files for each profile.
- State to go to - on or off

- Delay - seconds as a positive integer or 0. If the field is omitted 0 is assumed.

All outputs commence in the off state when Pi Presents starts. Commands in animate-begin are executed at the beginning of a track and animate-end at the end of a track. When the commands are executed the required changes to the GPIO are put in a queue for firing at the appropriate time. The time is not affected by pausing a track.

Animation commands with are not forgotten at the end of a track so animation can be synchronised over a mediashow. A side effect of this is that it is possible for a GPIO change to happen after a track or show has finished, and potentially after animate-end commands have taken effect. If you want to avoid this and ensure outputs are in a defined state at the end of a track set animate-clear to yes. The queue will then be cleared triggered, before animate-end is executed, of those events that were commanded by the track but not triggered.

8 Black Box Operation

There are a number of things to set up to make Pi Presents into a full screen, auto starting, GPIO controlled application. You do not need to use them all.

8.1 Command Line Options

`python pipresents.py -h` will show the command line options

Options	
-p --profile <arg>	Base name of the profile to be used. e.g. pp_mediashow If this is not specified then it defaults to pp_profile.
-g --gpio	Use the GPIO for buttons, triggers and animation. To use this Pi Presents must be run as root: `sudo python pipresents.py -g` The keyboard can still be used if gpio is enabled.
-b --noblank	Disable screen blanking. For this to function x11-server-utils must have been installed.
-f --fullscreen	Run Pi Presents in full screen mode.
-o --home <arg>	Location of the Pi Presents 'home' directory' e.g. /media/USBSTICK or /home/pi/my_data If this option is not specified it defaults to the users home directory. **NOTE**: If sudo is used then the user's home directory is /root so the option must be specified.
-d --debug	Run time errors give alerts and are reported to the terminal

	window if open. Turning on debugging will provide an additional trace of the operation of Pi Presents in the terminal window. Errors and the trace are also recorded in the /pipresents/pp_log.log file.
-v --validate	Validate the profile when Pi Presents is started.
-l --liveshow <arg>	Liveshow tracks are always played from the directory pp_live_tracks in the data home specified in the -o option defaulting to /home/pi/pp_home/pp_live_tracks If the -l option contains a directory path this is used as a second source for liveshow tracks. e.g. /home/pi/mylivetracks

8.2 *Specify a Profile*

The --profile (-p) command line option specifies the profile to use. This is the name of the profile directory in the pp_profiles directory. If the option is omitted it will default to pp_profile.

The prefix pp_ means nothing special other than denoting files provided by the developer.

8.3 *Specify a Home Directory*

The --home (-o) command line option specifies the location of Pi Presents data home. If the option is omitted it will default to the users home directory. **NOTE**: If sudo is used then the user's home directory is /root so the option must be specified.

Hence if both --profile and --home options are omitted the profile directory will be:

```
/home/pi/pp_home/pp_profiles/pp_profile
```

which should contain at least a pp_showlist.json file and optionally xxx.json medialist files. Media files should ideally be stored under:

```
/home/pi/pp_home
```

sub-directories are permissible.

USB sticks can be used to hold media. They will be assigned mount points in the /media directory.

e.g. /media/KINGSTON

Raspbian will auto mount USB sticks if they are present at power on, or plugged in afterwards. If Pi Presents is started at power on it takes up to 10 seconds after Pi Presents starts for the drive to be mounted.

Raspbian will compute the mount point from the name of the drive on the stick. If preparing the USB Stick on Windows machines it appears Windows converts all entered drive names to upper case.

8.4 Using GPIO to Control Pi Presents

GPIO control is enabled using the --gpio command option. If GPIO is used then the command starting Pi Presents must be preceded by sudo.

In the default configuration the following Pins of the GPIO connector 1 are bound to the following functions. The binding can be modified as described in Section 9.1

Default Pin	**Function**	**Equivalent Key**	
P1-12	shutdown	-	Press for 5 seconds to exit Pi Presents and shut down the Pi.
P1-15	down	Cursor Down	Next in show or down for menu
P1-16	up	Cursor Up	Previous in show or up for menu
P1-18	play	Return	Start a show or play a child show and images
P1-22	pause	p or spacebar	toggle pause for videos.
P1-7	stop	Escape	Stop playing a track or return to the higher level show.
P1-11	PIR	Return	Start a show.
-	-	CTRL-BREAK	Abort Pi Presents. Focus must be on a Pi Presents window.

The GPIO ports are set up as edge triggered inputs with internal pull-up resistors.

- Push buttons should be mechanical, push to make (normally open), and be connected between the GPIO pin and Ground.
- PIR's should have Normally Closed relay contacts and be connected between the GPIO pin and ground.

Inputs can be changed from normally open to normally closed and vice versa by changing the edge which is used for triggering as described in Section 9.1

A 330 ohm resistor is series with the button or PIR is recommended to protect the Pi should the inputs inadvertently be used as outputs.

GPIO Pin ----- 330R ----- contact ---- 0 volts (Gnd)

There is software contact de-bouncing which is set with a small hysteresis. If you have problems with contact bounce increase the THRESHOLD of the appropriate pin by modifying gpio.cfg (See Section 9.1).

8.5 Disable Screen Blanking

To disable screen blanking you must first install xset which is part of the x server utilities package, this should have done if you followed the install instructions..

```
sudo apt-get install x11-xserver-utils
```

You can than use the --noblank command option to disable screen blanking.

8.6 Start Pi Presents when Power is applied to the Pi

This will function only if you have set 'boot to desktop' using raspi-config.

- Create the folder /home/pi/.config/lxsession/LXDE

  Note: The directory .config is already present in the image but you will need to select 'Show Hidden Files' in the File Manager to see it.

- In this folder put a file named `autostart` containing one line specifying the full path:

  e.g. `sudo python /home/pi/pipresents/pipresents.py <options>`

- Make the autostart file executable by using the properties menu in the File Manager to alter the permissions on the file.

8.7 Shutdown the Raspberry Pi from the GPIO

Press the Shutdown button for more than 5 seconds. Pi Presents will exit and safely shut down the Pi.

9 Configuring Pi Presents

9.1 Configuring GPIO Inputs and Outputs

BEWARE: Accidentally using a pin as an output with the output shorted to ground or +3.3 volts will break your Pi, always use a series resistor on every input and output.

The configuration of the GPIO used by Pi Presents is defined in the file /pipresents/pp_home/gpio.cfg.

The default configuration allows for the buttons and PIR described in this manual. If you wish to change the configuration then copy the file to a profile directory, e.g. /home/pi/pp_home/pp_profiles/myprofile. Pi Presents will look for 'gpio.cfg' here before looking inside /pipresents. If changing the file beware:

- This file in /pipresents/pp_home may be overwritten by updates.
- The fields defined in this file may change when Pi Presents is updated.
- There is little checking of the content of this file by Pi Presents.

gpio.cfg is a text file which can be edited using Leafpad.

The gpio.cfg file contains information on the meaning of the fields. The configuration is split into sections, one for each available GPIO pin on P1. If a pin is not to be used then its section with a blank name must be left in the file.

Pin used as Input:
```
[P1-08]              # pin number of P1 connector.
name =  myinput      # a symbolic name used by Pi Presents or in medialists or
show   profiles.
direction =  in      # always in for inputs
threshold = 2        # the de-bounce threshold
front-edge = true    # 'true' if a callback is required on the front edge
back-edge =          # true if a callback is required on the back edge
```

Pin used as Output:
```
[P1-11]              # pin number of P1 connector
name =  myoutput     # a symbolic name used by Pi Presents in medialists or
show profiles
direction =  out     # always out for outputs
```

Unused Pin:
```
[P1-13]              # pin number of P1 connector
name =
```

Pi Presents uses the following symbolic names for the Buttons. If there is no pin with the appropriate name the input will not work:

play, pause, stop, up, down, PIR, shutdown

If not required these pins can be re-assigned to inputs for use as additional triggers for mediashows that are not 'controlled' or as outputs for animation.

9.2 Modifying on Screen Messages

A few administrative messages seen by customers cannot be controlled using profiles. The content of these messages is defined in the file /pipresents/pp_home/resources.cfg.

If you wish to change the content of these messages then copy the file to the data home directory, e.g. /home/pi/pp_home. Pi Presents will look for 'resources.cfg' here before looking inside /pipresents. If changing these messages beware:

- This file in /pipresents/pp_home will be overwritten by updates.
- The fields defined in this file may change when Pi Presents is updated.
- There is little checking of the content of this file by Pi Presents.

resources.cfg is a text file which can be edited using Leafpad. It has a section for each show or player which has messages which are displayed to customers. e.g.

```
[mediashow]
# waiting for user to start a show with progress = manual
m01 = To start the show press 'Play'

# waiting for trigger from PIR
m02 =
```

There is a comment (#.....) which describes how the message is used and a number, e.g. m01, which is used to refer to the message in the python code.

You can edit the text after the = sign to change a message or, if blank, to not display the message.

Make sure to try out your edits by running Pi Presents from the terminal window so that any Python Exceptions due to mistakes in edits are displayed.

Two messages allow the time of the next show to be displayed. This is achieved by inserting %tt in the message string.

10 Details of Profile Data

Out of the box configuration data is kept is a directory called /pp_home. /pp_home must contain a directory, /pp_profiles, in which profiles are stored.

A profile is a directory which contains the information needed to configure a single application of Pi Presents. It must contain a pp_showlist.json file and one or more medialist (.json) files.

For portability reasons all media should also be kept under the directory /pp_home, but this is not essential. Media can be stored anywhere under the /pp_home directory using sub-directories if necessary.

The default location of /pp_home is /home/pi; this will need to be overridden by command line options if you are using a USB stick. There is a /pp_home directory in the /pipresents directory, but this should not be used as it contains internal data.

The pp_showlist.json file contains a number of sections, a 'start' show section and a number of sections each defining a user generated show.

Shows in the pp_showlist.json file define the look and feel of the shows and how they link together. Medialist (.json) files in a profile define the content each show and some per track look and feel information.

10.1 Shows

10.1.1 Start Show

This must be present and serves just to initiate the first user defined shows.

Field	Examples	Values
type	start	A special type
title	First Show	Text describing the show displayed in the editor.
show-ref	start	must be 'start'
start-show	show1 show2	A list of 'show-ref''s separated by spaces.
controlled-show	show1	The show that will respond to button and key presses.

10.1.2 Mediashow

A 'Media show' is a sequence of one or more images, videos, messages, menus, mediashows or soundtracks. The content is defined by a medialist (.json) file. A show in the pp_showlist.json defines the look and feel of the show.

Field	Examples	Values
		Essential information
type	mediashow	Cannot be edited
title	My First Show	Text describing the show displayed in the editor or menus.
show-ref	show1	A ‘label’ by which the show is referenced by other shows. Any text without spaces
medialist	mymedia.json	Filename of the medialist file containing the tracks for the mediashow.
		How the show is to be run
disable-controls	no	yes/no If 'yes' keyboard and buttons are disabled for this show and any sub/child shows

Field	Examples	Values
trigger	start	How the media show proceeds after it is started: • start - continue without a wait • GPIO - wait for a trigger from a GPIO input • time/time-quiet - wait for a time of day. 'time-quiet' suppresses the next show message. • button - wait for the Play button or Return key
trigger-input	PIR	If trigger is 'GPIO': A symbolic name that is bound to a GPIO pin as described in Section 9.1 If trigger is 'time': A list of times of day h:m:s in 24 hour format separated by spaces. (See Section 5.1)
repeat	interval	How the media show is repeated: • oneshot - mediashow is run once • interval - run at intervals Reaching the repeat interval will not interrupt playing tracks so all tracks in a repeating show must have non-zero duration or end naturally. To have a blank screen at the end of a mediashow include a 1 second long blank message track.
repeat-interval	10	The time between the start of one mediashow and the start of the next in seconds: • If the value is less than the length of the show play will be continuous. • A value of 0 will always give a continuous repeat of the show
progress	auto	How the show progresses between tracks: • manual - Pi presents will wait for next/previous keys/buttons between tracks. • auto - tracks play continually, next/previous keys/buttons can be used to skip.
next-trigger	button	If progress is manual how the mediashow proceeds to the next track: • GPIO - wait for a trigger from a GPIO input • none - next-trigger is inoperative
next-input		A symbolic name that is bound to a GPIO pin as described in Section 9.1 next-trigger = GPIO its input will be used to move to the next track.

Field	Examples	Values
sequence		sequence of tracks: • ordered - played in the order of the medialist • shuffle - play tracks in a random order There is no natural end to a shuffled set of tracks so it will be necessary to use trigger-end if the show is to be terminated.
trigger-end		none • the end trigger is not operative time • end the show at a time of day duration • end the show after a period of time.
trigger-end-time	5:00	If trigger-end is 'time': • A list of times of day h:m:s in 24 hour format separated by spaces. (See Section 5.1) If trigger-end is duration • a period of time in h:m:s format.
		Mediashows can optionally provide an entry point for a menu. If the Return key or Play button is pressed the menu will be started. The mediashow will continue when the menu terminates.
has-child	yes	yes/no. If yes the child is enabled. This is usually a menu but could be a mediashow. The child is defined in the medialist with the track-ref of pp-child-show.
		A hint is a line of text displayed near the bottom of the screen when has-child is yes. Used as a hint on the existence of a menu and how to enter it.
hint-text	Press Play to..	
hint-font	Helvetica 30 bold	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
hint-colour	white	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
hint-y	100	distance of the text from the bottom of the screen (pixels)
		Show text is overlayed on all images in the show.
show-text	Picture of Taj Mahal	If not blank the text displayed on the image.

Field	Examples	Values
show-text-font	Helvetica 30 bold	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
show-text-colour	white	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
show-text-x	100	distance of the start of the text from the left of the screen (pixels)
show-text-y	100	distance of the top of the text from the top of the screen (pixels)
		Tracks played in the show need some configuration if not supplied in the medialist
transition	cut	cut, fade, slide, crop. Type of transition between tracks. Only cut works until Pi Presents gets GPU acceleration. Others are implemented as an experiment, but incorrectly.
duration	10	How long an image or message is displayed in seconds. A value of 0 displays continuously.
omx-audio	hdmi	hdmi/local/<blank>. Sound output channel for any video track played by omxplayer from the show. If blank then the channel is set automatically by the presence of a hdmi monitor.
omx-volume	0	Volume of video track (-60 - 0 dB) played by the show. May be overridden by the track.
omx-other-options		other options for omxplayer (care required to avoid have a nice day!)
mplayer-audio	local	hdmi/local. Sound output channel for any audio track played by mplayer from the show.
audio-speaker	stereo	left/right/stereo. Speaker for any audio track played by mplayer from the show.
mplayer-volume	0	Volume of audio track (-60 - 0dB) played by the show. May be overridden by the track.
mplayer-other-options		Other options for mplayer (care required to avoid rejection!)

10.1.3 Menu

A 'Menu' is a collection of one or more images, videos, messages, menus, mediashows or soundtracks. The content is defined by a medialist (.json) file. A show in the pp_showlist.json defines the look and feel of the menu.

Field	Examples	Values
		Essential information
type	menu	Cannot be edited
title		Text describing the show displayed in the editor or menus.

show-ref	mymenu	A ‘label’ by which the show is referenced by other shows. Any text without spaces
medialist	themenu.json	Filename of the medialist file containing the tracks for the menu.
		The look and feel of the menu
disable-controls	no	yes/no If 'yes' keyboard and buttons are disabled for this show and any sub/child shows
menu-x	300	integer, x position of the start of the first line of the menu (pixels from left of screen)
menu-y	250	integer, y position of the start of the first line of the menu (pixels from top of screen)
menu-spacing	70	integer, gap between menu entries (top to top)
entry-font	Helvetica 30 bold	Tkinter font name see: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
entry-colour	white	Tkinter colour see: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
entry-select-colour	red	colour when the entry selected
timeout	60	seconds, If there is no activity on the menu Pi presents automatically returns to the previous display. 0 for no timeout.
has-background	yes	yes/no. If yes an image is displayed behind the menu. The image is defined in the medialist with the track-ref of pp-menu-background.
		Menus do not have children; the hint is used for track playing instructions.
hint-text	To Play..	
hint-font	helvetica 30 bold	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
hint-colour	white	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
hint-y	100	distance of the text from the bottom of the screen (pixels)
		Show text is overlayed on all images opened from the menu.
show-text	Picture of Taj Mahal	If not blank the text displayed on the image.
show-text-font	Helvetica 30 bold	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
show-text-colour	white	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
show-text-x	100	distance of the start of the text from the left of the screen (pixels)

Field	Examples	Values
		Essential information
type	menu	Cannot be edited
title		Text describing the show displayed in the editor or menus.
show-ref	mymenu	A 'label' by which the show is referenced by other shows. Any text without spaces
medialist	themenu.json	Filename of the medialist file containing the tracks for the menu.
		The look and feel of the menu
show-text-y	100	distance of the top of the text from the top of the screen (pixels)
		Tracks played from a menu need some configuration if not supplied in the medialist
transition	cut	see mediashow for values
duration	10	see mediashow for values
omx-audio	hdmi	see mediashow for values
omx-volume	0	Volume of video track (-60 - 0 dB) played by the show. May be overridden by the track.
omx-other-options	-t 1	see mediashow for values
mplayer-audio	local	hdmi/local. Sound output channel for any audio track played by mplayer from the show.
audio-speaker	stereo	left/right/stereo. Speaker for any audio track played by mplayer from the show.
mplayer-volume	0	Volume of audio track (-60 - 0 dB) played by the show. May be overridden by the track.
mplayer-other-options		other options for mplayer (care required to avoid rejection!)

10.1.4 Liveshow

A 'Live show' is a sequence of one or more images, videos and soundtracks. The content is dynamically supplied. A show in the pp_showlist.json defines the look and feel of the show.

Field	Examples	Values
		Essential information
type	liveshow	Cannot be edited
title	My Live Show	Text describing the show displayed in the editor or menus.
show-ref	myliveshow	A 'label' by which the show is referenced by other shows. Any text without spaces

Field	Examples	Values
disable-controls	no	yes/no If 'yes' keyboard and buttons are disabled for this show and any sub/child shows
trigger-start	time	start/time How the liveshow proceeds after it is started: • start - continue without a wait • time/time-quiet - wait for a time of day. 'time-quiet' suppresses the next show message.
trigger-start-time	9:30 2:30	A list of times of day h:m:s in 24 hour format separated by spaces. (See section 5.1)
trigger-end	none	none/time Trigger the end of the show • none - continue forever • time - terminate at a time of day
trigger-end-time	12:00 17:30 5	If trigger-end is 'time': • A list of times of day h:m:s in 24 hour format separated by spaces. (See Section 5.1) If trigger-end is duration • a period of time in h:m:s format.
medialist	mymedia.json	Filename of the medialist file containing the child-show track for the liveshow. Must be present even if there is no child-show.
		Liveshows can optionally provide an entry point for a menu. If the Return key or Play button is pressed the menu will be started. The liveshow will continue when the menu terminates.
has-child	yes	yes/no. If yes the child is enabled. This is usually a menu but could be any type of show. The child is defined in the medialist with the track-ref of pp-child-show.
		A hint is a line of text displayed near the bottom of the screen when has-child is yes. Used as a hint on the existence of a menu and how to enter it.
hint-text	Press Play to..	
hint-font	helvetica 30 bold	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
hint-colour	white	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
hint-y	100	distance of the text from the bottom of the screen (pixels)

Field	Examples	Values
		Show text is overlaid on all images in the show.
show-text	Picture of Taj Mahal	If not blank the text displayed on the image.
show-text-font	Helvetica 30 bold	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
show-text-colour	white	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
show-text-x	100	distance of the start of the text from the left of the screen (pixels)
show-text-y	100	distance of the top of the text from the top of the screen (pixels)
		Tracks played in the show need some configuration.
transition	cut	cut, fade, slide, crop. Type of transition between tracks. Only cut works until Pi Presents gets GPU acceleration. Others are implemented as an experiment, but incorrectly.
duration	10	How long an image is displayed in seconds. A value of 0 displays continuously.
omx-audio	hdmi	hdmi/local/<blank>. Sound output channel for any track played by omxplayer from the show. If blank then the channel is set automatically by the presence of a hdmi monitor.
omx-volume	0	Volume of video track (-60 - 0 dB) played by the show. May be overridden by the track.
omx-loop	no	seamless loop of tracks in liveshow. (Not implemented.)
omx-other-options		other options for omxplayer (care required to avoid have a nice day!)
mplayer-audio	local	hdmi/local. Sound output channel for any audio track played by mplayer from the show.
audio-speaker	stereo	left/right/stereo. Speaker for any audio track played by mplayer from the show.
mplayer-volume	-10	Volume of audio track (-60 - 0 dB) played by the show. may be elvetica by the track. If blank then the volume is not changed.
mplayer-other-options		other options for mplayer (care required to avoid rejection!)

10.2 Medialists

Medialists define the content of a show. Each entry is a 'track'. Tracks can be of various types:

- image - a still image. Image types are currently those that can be rendered by the Python Imaging Library. Image size should be kept to the 1 megapixel region. Images can be overlaid with Track Text and Show Text. Animation is facilitated.

- video - a track played by omxplayer. Track types depend on the codec licences purchased from the Foundation. Animation is facilitated.

- audio - an audio track played by mplayer. Any track type playable by mplayer should be playable. The audio track type is very flexible as all elements of it are optional and it has extended duration handling in addition to animation facilities.

- message - displays lines of text against a black background. Can also used to display a blank screen. It provides a simple slide preparation facility. If you want something better use Powerpoint or similar and export as .jpg displaying as an image.

  Note: The message player is an early development which has minimal flexibility and is now replaced by the audio player (but retained for backwards compatibility). To display text it might be better to use 'Track Text' on an 'audio track' with a blank audio file and a non-zero duration.

- show - shows can be used as tracks allowing nesting to any depth.

- child-show - a special type of track to specify the child show of a mediashow.

- menu-background – a special type of track to specify the image file to be used as the background for a menu.

Media Track Path References
The leading + sign in file paths allows tracks to be specified relative to the /pp_home directory. If a plus sign is not present then the path must be absolute. It is recommended that media files be stored under /pp_home if the profiles are to be portable.

Track Reference Labels
Some medialist entries will have labels defined by the key 'track-ref'. If the label is blank then the track is included in the menu or mediashow. If the label is not blank then the track is used for a special purpose. Currently there are two special purposes;

- The background image for a menu which has the label pp-menu-background

- The child show of a mediashow which has the label pp-child-show

Labels will be used in the future to respond to asynchronous events.

The field definitions for each type of medialist track follow.

10.2.1 Image

Field	Example	Values
track-ref		A label for the track - blank so that the track is included in a menu or mediashow.
type	image	
title		Displayed on a menu and in the editor
location	+/media/sarename.gif	The filename of the track which may be blank.
duration	5	Seconds. If 0 then image is displayed until terminated by user input. If blank then the value in the referencing show is used.
transition	cut	cut/fade/rotate/slide/. If omitted then the value in the referencing show is used. (Only cut currently works)
track-text	Picture of Taj Mahal	If not blank the text displayed on the image.
track-text-font	elvetica 30 bold	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
track-text-colour	white	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
track-text-x	100	distance of the left end of the text from the left of the screen (pixels)
track-text-y	100	distance of the top of the text from the top of the screen (pixels)
animate-begin	out1 on 1 out1 off 2 out2 on 1 out3 on 6	See Section 7
animate-clear	no	yes/no See Section 7
animate-end	out2 off	See Section 7

10.2.2 Video

A track to be played with OMXPlayer.

Field	Example	Values
track-ref		A label for the track - blank so that the track is included in a menu or mediashow.

type	video	
title	The Film	Displayed on a menu and in the editor
location	/home/pi/myvideos/film.mp4	The filename of the track. (absolute)
omx-audio	local	hdmi/local. If blank then the value in the referencing show is used.
omx-volume	0	Volume of video track (-60 - 0 dB) played by the show. May be overridden by the track.
omx-window	10 10 500 1000	window for displaying video tracks x1 y1 x2 y2 (not used)
omx-loop	no	Seamless looping of tracks. Not implemented.
track-text	Picture of Taj Mahal	If not blank the text displayed on the image.
track-text-font	helvetica 30 bold	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
track-text-colour	white	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
track-text-x	100	distance of the left end of the text from the left of the screen (pixels)
track-text-y	100	distance of the top of the text from the top of the screen (pixels)
animate-begin	out1 on 1 out1 off 2 out2 on 1 out3 on 6	See Section 7
animate-clear	no	yes/no See Section 7
animate-end	out2 off	See Section 7

10.2.3 Audio

An audio track to be played with Mplayer. The audio 'track' is very flexible and designed for use as the basis for controlling animated exhibits.

Field	Example	Values
track-ref		A label for the track - blank so that the track is included in a menu or mediashow.
type	audio	
title	The Music	Displayed on a menu and in the editor
location	+/tracks/music.mp3	The filename of the track. The location is optional. If blank the timing of the 'track' is taken from the duration field which may not be blank.?????
clear-screen	no	yes/no The screen is normally not cleared at the start of an audio track as this would clear

		any concurrent displays from other shows.
background-image	+/images/back.jpg	The filename of an optional image that is displayed on the background, can be blank. Show and Track text is displayed above the image.
duration	5	Seconds. If duration is 0 then the track ends when the audio file ends or is aborted, or never ends if there is no audio track. If non-zero the 'track' ends at the stated time. The playing of the track may be cut short or there may be period of silence after the audio file ends if duration is non-zero.
mplayer-audio	local	hdmi/local. If blank then the value in the referencing show is used.
audio-speaker	left	left/right/stereo. If blank then the value in the referencing show is used.
mplayer-volume	0	Volume of audio track (-60 - 0 dB). If blank then the value in the referencing show is used.
track-text	Picture of Taj Mahal	If not blank the text displayed on the image.
track-text-font	helvetica 30 bold	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
track-text-colour	white	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
track-text-x	100	distance of the left end of the text from the left of the screen (pixels)
track-text-y	100	distance of the top of the text from the top of the screen (pixels)
animate-begin	out1 on 1 out1 off 2 out2 on 1 out3 on 6	See Section 7
animate-clear	no	yes/no See Section 7
animate-end	out2 off	See Section 7

10.2.4 Message

Field	Example	Values
track-ref		A label for the track - blank so that the track is included in a menu or mediashow.
type	message	
title	A Message	Displayed on a menu and in the editor
text	Welcome	The text to be displayed.

duration	5	Seconds. If 0 then message is displayed until terminated by user input. If blank then the value in the referencing show is used.
message-font	helvetica 30 bold	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
message-colour	white	See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm
background-image	+/images/back.jpg	The filename of an optional image that is displayed instead of a plain background, can be blank. The message is displayed on top of the image.
background-colour	red	If not blank the colour of the background. See: http://effbot.org/tkinterbook/tkinter-widget-styling.htm

10.2.5 Show

Shows can be a track in another show or menu. Uses might be:

- A mediashow made up of smaller mediashows
- A menu of mediashows, particularly presentations
- Menus with sub-menus
- Liveshow run from a Mediashow which provides an initial screen.

Field	Example	Values
track-ref		A label for the track - blank so that the track is included in a menu or mediashow.
type	show	
title	My Other Show	Displayed on a menu and in the editor
sub-show	myothershow	Reference to the show to be run

10.2.6 Child Show

This is a special medialist entry which provides a referencing mechanism for the child show of a mediashow.

track-ref	pp-child-show	The label indicates this contains a reference to a child show. The track will not be included in the menu or mediashow.
type	show	
title	Child Show	Displayed on a menu and in the editor
sub-show	mychildshow	Reference to the show to be run

10.2.7 Menu Background

Field	Example	Values
track-ref	pp-menu-background	The label indicates this contains a reference to a menu background. The track will not be included in the menu or mediashow.
type	menu-background	
title	Menu Background	Displayed on a menu and in the editor
location	+/media/sarename.gif	The filename of the track. (relative)

11 Hardware Requirements

Pi Presents can be used with either Rev 1 or Rev 2 Pi's. The GPIO pins have been chosen such that they are version agnostic. 256MB Pi's can be used provided image size kept to the 1 megapixel region.

The RAM memory split is determined by omxplayer and on 256MB machines should be the minimum that omxplayer requires to allow the maximum RAM memory for the handling of images. omxplayer plays some videos using 64MB of RAM; others need 128MB, especially if you want sub-titles.

Display of images is slow. A one megapixel images takes a couple of seconds to display. The one 10 megapixel image I tried took 10 seconds to display and crashed a 256MB Pi. Larger images, greater than the screen pixel size, will do nothing to improve the picture and will take longer to display even on 512MB machines; I use the brilliant Faststone Photo Resizer http://www.faststone.org/FSResizerDetail.htm to reduce the size of images on a Windows PC. The todo list includes using the GPU to display images.

Use the HDMI or 3.5mm jack output for audio. Amplification and volume control will need to be provided in external hardware to suit the application.

12 Updating Pi Presents

Download pipresents from github as described in the README.md file. Data stored in /home/pi/pp_home will not be affected. Do not store your data in the pipresents/pp_home directory as it may be overwritten.

As Pi Presents develops new fields will be added to the profile definition and others deleted. To control this, the three elements - Pi Presents, the editor and the profiles - must have the same version. Pi Presents will object if a profile with the wrong version is used. Use the editor which will automatically update the version of the profile and its fields.

The profiles in pipresents-examples will be kept compatible with the latest version of Pi Presents. Beware, re-installing these might overwrite profiles you have made. The example profiles can be updated by running them through the editor.

When updating consider that you may need to update 'resources.cfg'

13 Bug Reports and Feature Requests

Please use the Github Issues Tab https://github.com/KenT2/pipresents/issues or the Pi Presents thread http://www.raspberrypi.org/phpBB3/viewtopic.php?f=38&t=29397on the Raspberry Pi forum to report bugs.

I am keen to improve Pi Presents and your input on real world experiences and requirements would be invaluable to me, both minor tweaks to the existing functionality and major improvements.

14 Known Problems

14.1 Pi Presents locks up in full screen, how do I escape.

This is usually caused by Python reporting an exception due to incorrect configuration data.

To avoid this use validation in the editor and try the show, not full screen, and running from a terminal window so you can CTRL-C out if CTRL-BREAK fails. If this fails, try closing the terminal window and if necessary reboot. Pulling the power has potential for corruption, so the ideal solution is to SSH into the Pi from another machine, run top (top -upi) and kill the python process with the k xxxx command. You may need to kill an omxplayer process as well.

14.2 Permission to write to pp_log.log is denied.

The log file pp_log.log is created with the user permissions currently employed. If sudo was used in the command that created the file it is owned by root. Delete the file with user root and then create with user Pi or run pipresents.py without sudo.

14.3 Pi Presents sometimes crashes when two audio tracks are being played

I have found that if two audio tracks are played with MPlayer and one is set to HDMI and the other to Local then as expected the sound output channel switches. It also causes MPlayer to crash. The command used to change is amixer -c 0 numid=3.

14.4 When using the editor on Windows I get the message "failed to save medialist, trying again "

When using Windows the medialist is not reliably saved the first time. However if you try again it will probably succeed. Ideas on why this happens welcome. Code is in pp_medialist.py at around line 266.

14.5 Pi Presents crashes after playing videos for a few hours

There seems to be some timing problem in the interface with OMXPlayer or with OMXPlayer itself. If a crash happens its usually best to restart the Raspberry Pi as OMXPlayer seems to be left in a funny state, see Github Issue ?? for more information