OmxplayerGUI Manual

Version 1.7.0

Content

1) Introduction

omxplayerGUI is a program to manage playing all kinds of media with omxplayer in lots of different ways. It's one of the helper programs of the kweb package (Minimal Kiosk Browser), but it can also be used standalone. omxplayerGUI also supports extracting videos from websites containing HTML5 video or from all video websites supported by youtube-dl. It uses Python's Tkinter GUI elements, which look a bit simplistic by modern standards, but do not require any other packages to be installed. OmxplayerGUI has been written in Python and then compiled to a binary version using nuitka.

Omxplayer is the most efficient media player for the Raspberry Pi. It's a command line program which is controlled by keyboard commands and can also be controlled from other applications using dbus commands. OmxplayerGUI uses both methods to control omxplayer.

It's important to understand how omxplayer works. It uses the GPU of the Raspberry Pi to decode the video and sends it directly to the video output superimposing it on top of the normal display (of the desktop, for example) in a separate layer. That's one of the reasons, why it's so efficient. Other methods can be used to display the decoded video inside a GUI application, but this requires real time copying of the decoded video into the frame buffer; the webkit3 engine supplied by the Raspberry Pi Foundation (and used by epiphany and kweb3) works this way to play (HTML5) web video inside the browser window. But this method is very inefficient compared to omxplayer's layer method and only works well for low resolutions and limited screen areas.

OmxplayerGUI works differently: It looks like it's playing the video inside a window, but in fact the video area is still running in a screen overlay; you will immediately notice that, if you move another window on top of the video window: the video is shining through. But nevertheless you can move or resize the video window and the video size and position will immediately match the new window size and position without breaking the video playback. You can also hide the video window and let it play on in the background. Full screen and refresh modes (adjusting the video display to the size and frame rate of the video) are of course also supported.

This can only work if omxplayer and the GUI use the same coordinate system and screen size. All kinds of overscan settings will break this, because the frame buffer size (used by the GUI environment) and the real screen size (used by omxplayer) don't match. I'm quite sure, that overscan is not needed in 99.99% of all cases; modifying the settings of your monitor or TV or using custom resolutions (only in rare cases) instead is always a better solution.

OmxplayerGUI is used by kweb(3) to play local or web video of all kinds, but it can also be used standalone as a desktop video player. It registers with the desktop environment as a separate media player application and can be opened from the application menu or from the file manager, using its “open” or “open with” commands for all kinds of media files or playlists. If it is called without any arguments (from the application menu, for example) it will display a simple frontend window. In version 1.7 this frontend has been expanded to help you creating and managing your own playlists.

2) Installation and Usage

omxplayerGUI is installed as part of the Minimal Kiosk Browser (kweb) package. A separate installation is not supported any more, because kweb and omxplayerGUI share a common environment (helper programs and global settings). You might think of omxplayerGUI as an application that has two interfaces: kweb(3) and a standalone media player interface.
So you have to install kweb even if you are only interested in the media player itself. You don't have to use the browser if you don't want to, but it is at least required to manage omxplayerGUI's rather complex settings.

After installation you will find omxplayerGUI as a new program in your applications menu. From a terminal command line you can run it with:

omxplayergui [option] [mode] [url] [mimetype]

“option” may be one of these:
“--preset=presetname” or “--config=fullpath”,
where “presetname” must be the name of a preset file created with kweb's settings page and “fullpath” must be the full path to a python file (.py), that contains global variables, similar to kwebhelper_settings.py

“mode” may be either 'av' (or -av) or 'ytdl' (or -ytdl) and url is any valid file path or web url. If no mode is given, 'av' will be used. 'ytdl' is the extraction mode for web video and requires a valid URL to a web page containing web video. OmxplayerGUI uses youtube-dl to extract video links from websites like youtube.com and many others. It can also extract HTML5 video links directly.

You can also right click any media (audio, video or playlist) file inside your file manager window to open it with omxplayerGUI.

If no argument is given or called from the application menu, omxplayerGUI will start with a simple frontend.

For web video support, you have to install youtube-dl. This is a very powerful Python application which allows you to download web video from a lot of websites (more than 600). OmxplayerGUI uses it to extract web video URLs from web pages and feeds those video URLs directly to omxplayer for immediate playback (nothing is downloaded!).

Unfortunately the youtube-dl version from the Raspbian repository is useless, because it cannot be updated. The video websites are constantly changing and youtube-dl has always to adapt to those changes. There are different methods to install and update youtube-dl (and all of them work with omxplayerGUI), but to get the best performance you should use the tools provided by the kweb environment: Open kweb's applications page. At the bottom you'll find a number of youtube-dl tools (buttons). Click the “Install (Git)” buttons to install a recent version from github. Use the button “Update (Git)” from time to time to get the newest version.

To speed up the use of youtube-dl (access time) version 1.7.0 of the kweb package now contains a youtube-dl-application-server. This server can also be started (and closed) from a button on the application page. OmxplayerGUI's standalone frontend will start the server automatically, if the server is available and not already running.

3) omxplayerGUI frontend

When not used from Minimal Kiosk Browser and started without any arguments, omxplayerGUI will open with a simple frontend:

The interface consists of an entry line (for file paths or URLs), a first row of action buttons, a playlist area and a second row of action buttons to manage playlist content.

If you click the "**Open**" button, a simple file dialogue will appear and you may select any kind of media or playlist file to play. Depending on the kind of file, either omxaudioplayer or omxplayerGUI (the video player) will open and start playing your file(s).

The "**Play/Stream**" and "**Extract**" functions require a valid URL or file path, that you have to enter in the text field above. You can use copy & paste for this, by right clicking into the text field and selecting "clear & paste" from the small pop-up menu.

If you have a valid (full) file path or a web link URL pointing to an audio/video file or stream on the internet (or a playlist) entered into the text field, you can play it by clicking on "**Play/Stream**". Depending on the content, either the audio or video player interface will be opened.

If you have entered a URL pointing to a web video page (something like "https://www.youtube.com/watch?v=o774LMtfeJY"), you can click the "**Extract**" button. Youtube-dl will be used to extract the video URL from that link and, if successful, omxplayerGUI will open and start playing the video. You can also use links to web pages containing HTML5 video and then omxplayerGUI will extract the video URLs from such pages and play them.

There's also a special function included to search for videos on youtube.com. If you enter something like:

```
?raspberry pi
```

into the text entry field and click on "Extract", a search on youtube.com for these key words will be started and after a while (around one minute!), a video window will appear and start playing the first video of the search results. If you stop the video, you will see a playlist with up to 20 search results for "raspberry pi", all of which can be directly played without any further delay. If the last of the search terms is a number, it will be used to set the page number of the youtube search, e .g.

```
?raspberry pi 3
```

will result in page3 (videos number 40 - 59) of a youtube search for "raspberry pi."

The "**Help**" button can be used to open this manual directly from the player.

If you click the "**Edit Settings**" button it, it will open a browser window as root (password may be required) and show the settings page for kwebhelper and omxplayerGUI (see below for details).

Clicking the "**Quit**" button will close the application, but only if all player windows have been closed.

The lower range of buttons is used for creating and managing playlists:

"**Load**" will open a file selector and you can select any m3u(8) or pls playlist from your file system. The contents will be shown in the playlist area above.

Clicking "**Save List**" will let you save a previously created or modified playlist in M3U format.

"**Add Files**" will open a file selector and you can add one or more media files to your playlist. To select multiple files you have to use the SHIFT and CTRL keys in the usual way.

When you click the "**Folder**" button, you can select a directory containing media files. Media files from sub-directories of all levels will also be included.

The "**Delete**" button will let you delete one or more selected entries from your playlist. Multiple entries can be selected by drawing the mouse with the left mouse button held down or by using the SHIFT and CTRL key in the usual way.

Click "**Play All**" to open your complete playlist in a player window and start playing. Depending on the content it will open an omxaudioplayer (audio only) or an omxplayerGUI (video or mixed content) window.

To play a single file from your playlist simply double click its name entry.

4) omxaudioplayer

If you open an audio file or a playlist (m3u and pls) containing only audio files or streams, omxplayerGUI will be opened in audio mode as “omxaudioplayer”. All files will be shown in a playlist.

Buttons and their keyboard controls:

Play/Pause – Space, p, Return or Keypad Enter: will play the currently selected song, or pause or resume playing, when it is already playing.
Stop – ESC or q: will stop playing a song.
Backward - ←: Jump backwards by about 10 seconds.
Forward - →: Jump forward by about 10 seconds.
Previous – ↑: If a song is playing and selected, stop it and play the previous song in the list. If a song is playing and you have selected another song, jump to that song and play it. If no song is playing, select the previous song and play it.
Next – ↓: If a song is playing and selected, stop it and play the next song in the list. If a song is playing and you have selected another song, jump to that song and play it. If no song is playing, select the next song and play it.

Volume control: move the slider to change the volume between -60 and +12 db. You can also click into the grey areas besides the slider to change the volume by 3 db, or use the '-' and '+' keys (also on the numeric keypad).

The keyboard commands are very similar to those used by omxplayer, except for the Up and Down arrow keys, which are not used for jumping ahead or backwards by 10 minutes, but for playing the previous or next song.

Functions inside the playlist window:

If you select another song, while a song is playing, it will be played when the current song is finished or when you click on the Previous or Next button. If you double click another song than the currently selected one, it will be played (and a currently playing song will be stopped). If you double click the currently playing song, it will be stopped.

5) omxplayerGUI – the video player window

a) Understanding the two different modes

Depending on a setting in kwebhelper_settings.py (edited via kweb's settings page), two different modes can be used.
freeze_window = False = extended mode, or
freeze_window = True = simple mode

In simple mode (without using dbus commands) a video window is "frozen" to the screen while a video is playing. The video window cannot be moved, resized or minimized until the video has been stopped. The use of simple mode is shown in the title bar of the window with an extension: "(f)" (= frozen).

In extended mode the video window can be moved, resized or minimized while a video is playing. The video area will be adjusted as soon as your moving or resizing action is finished. While the window is minimized (hidden), the video will be invisible but play on and you will still hear the sound.

Note: Using the simple mode is not recommended any more. It was a workaround for older omxplayer versions at the time omxplayerGUI was first published.

Even in extended mode, the video window is “frozen” for a few seconds at the beginning (shown in the title bar) and some controls are disabled, while the program checks, if seeking in the video is possible and if its duration is known. If seeking is not possible, the window remains in the frozen state. If a duration cannot be found, the seek bar remains disabled. It's also possible to toggle between both modes at run time with a keyboard command ALT+u (if video has been stopped).

Some videos (especially rtmp://, rtsp:// and mmsh:// streams) will always be played in frozen or simple mode, if they do not support seeking.

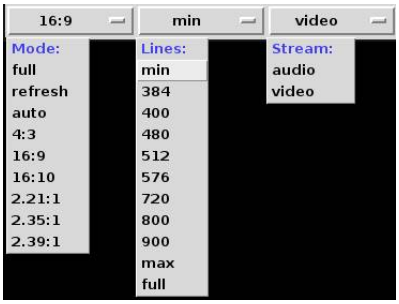
You can play more than one video at the same time. I've managed to play up to 4 videos at the same time, 3 SD video and one 720p video TV stream, or even 6 SD videos. This shows how powerful the GPU of the RPi is. If you play more than one video and pause one of them, the sound will get lost on all videos. You have to pause and start again each video to get the sound back (this may change with newer versions of omxplayer, see issue https://github.com/popcornmix/omxplayer/issues/184).

It's also possible to play one video on top of another one using different layers (see keyboard commands below). If multiple video windows are opened from the frontend window, they are layered automatically. This way it's very easy to create a PIP (picture in picture) effect.

b) Using the interface

When running in video mode, the GUI window has two lines of control elements at the bottom. The first line is the same as within an omxaudioplayer window (see above for details). The second line contains three menu buttons and another slider, which will be explained in detail here. In simple mode, you can only change these settings when no video is running; in extended mode most of these settings are available while a video is being played.

It's also possible, to hide these controls completely (and use only keyboard commands to control the player window). This can be toggled at run time with ALT+h. You can start the player in this mode by setting “hide_controls” to “True” (see below)



The "Mode:" menu contains the following choices:

full - play video full screen without any window elements

refresh - play video full screen without any window elements with omxplayer's -r option to change screen resolution and frequency

Both of these options only have an effect, when no video is playing. You have to select full screen mode, before you start a video.

Note: It's always possible to switch into a “simulated” full screen mode while a video is playing using two keyboard commands: ALT+h (to hide the controls), followed by ALT+f (to go to full screen). Both are toggle commands and by applying them again you'll return to window mode.

auto – If you select this mode before starting a video, the program will try to get information about the aspect ratio of the video and set the play area accordingly inside the current window. This will take another two or three seconds, before the video starts - on large avi files even more. This works both in simple and extended mode.

If the video has been started in one of the other numbered modes, auto detection of the aspect ratio always runs in the background and will be available after a short while; so you can switch to auto mode when the selected mode doesn't fit.

4:3, 16:9, 16:10, 2.21:1, 2.35:1 , 2.39:1

All these options define the ratio of width to height of a video manually. In extended mode, you can change these settings while a video is running. In simple mode only when a video has been stopped or not started yet. If none of these ratios seems to be right, choose auto mode (in extended mode you can also change this at run time).

Note: It depends on the omxplayerGUI settings if those modes stretch both the video area and the video window or just the window to match the video size. There's a “trueaspect” settings template

available, which will tell omxplayer to always use the correct video aspect (inside the window), independent of the window size (letterbox mode). This requires an omxplayer version from 2016.

The "Lines:" menu contains a number of settings that define the height of the video area, ranging from
"**min**" (minimal height, see below)
then some numbers like this:
320,384,432,480,512,576,720,864,960
to
max (maximize window)
and
full (full screen without window title bar, but controls at the bottom)

The "min" size depends on the font height used for the interface, defined on the settings page. The smallest height is 288 lines, when the font height is set to the minimal value of 10 (points). In all other cases it's font height*28. The numbers following "min" depend on the min value and on the size of your screen. In extended mode you can change the height of the video at run time and the video area will be resized immediately. It's also possible to resize the video window manually by dragging the corner of the window.

The "Stream:" menu contains only two options, "video" and "audio". It can only be changed if the player has been stopped. In some cases omxplayerGUI cannot decide if a stream link provides audio (e. g. web radio) or video. You can select whether the player should handle such streams as audio or video streams. It's not really important in most cases, as omxplayer will handle it by itself, but you can avoid getting a black screen (when full or refresh mode has been selected) when you just want to listen to a web radio stream.

If the player window is large or near the bottom of the screen, the menu list will pop up upwards and would be hidden behind the video area if a video is running. To avoid this, the video is set to half transparent for 2 seconds when you click any of the menu buttons. You can also control all these menus from the keyboard (see below for details).

The video position slider: before you start a video you can set the starting position (between 0 and 180 minutes, in steps of 3 seconds). In extended mode you can also set the video position at run time, but only if seeking in the file or stream is possible and a duration is known. This will be available a few seconds after starting the video, and it will also be limited to the real duration of the video.
The video position slider display units are minutes with seconds as decimal value (0.05 matches 3 seconds).

Note: The slider position is not updated to the actual video position while a video is running. This would require a rather heavy communication between omxplayer and omxplayerGUI and I've decided against it; I prefer things to be fast and snappy and using as few resources as possible.

c) Keyboard control

omxplayerGUI (and also omxaudioplayer) supports all keyboard commands of omxplayer with two exceptions:
The Up and Down arrow keys are used for playlist control (go to next or previous song or video in the playlist) instead for large jumps (600 seconds) forward or backward. You have to use PageUp

and PageDown instead (or “,” and “.”).
For some commands omxplayerGUI supports additional options:
Space bar, 'p', Return and Enter on the numeric pad all play/pause a video
'+' and '-' can also be used from the numeric keypad for sound volume control.
Appendix A contains a list of all keyboard commands incuding a comparison between omxplayer and omxplayerGUI.

omxplayerGUI supports its own special command set using the left ALT key in combination with the following keys:

c - close player (also in omxaudioplayer)

m - toggle Lines: mode between max and min (or toggle between maximize and un-maximize in audio player mode).

f - toggle Lines: mode between full and max (or toggle between full screen and window in audio player mode).

0 ... 9: select Lines: mode between "min" (0) and largest number. Depending on the number of available options, the larger numbers all may have the same effect.

'+' and '**-**' move through aspect ratios forward or backward (between 'auto' and '2.39:1')

a - .set ascpect ratio to 'auto'

u - toggle between simple and extended mode (visible in title bar where an '(f)' is added for simple mode).

s - save playlist as m3u file (also in omxaudioplayer).

h - hide / show all control elements at the bottom

k - the kill switch: sometimes omxplayer hangs and does not return. This command sends a killall command to all omxplayer.bin instances and also to all send-dbus instances.

NumPad+ and **NumPad-** or **PageUp** and **PageDown** - Set the video layer (default and minimum = 0) to a higher or lower value; this will allow overlays. Layer numbers > 0 will be shown in the title bar. Only possible when no video is playing.

If you stop playing a video file, the omxplayerGUI window will switch to playlist view. Handling the playlist is done exactly the same way as in omxaudioplayer (see above).

d) Windowless Mode

omxplayerGUI can also play videos in a windowless mode (depending on special settings, see below). In this case a terminal (xterm) will be started, to blank and fill the whole screen and give keyboard control and then play videos full screen with omxplayer. This is the old mode used by Minimal Kiosk Browser up to version 1.4. This mode is not used from the standalone frontend.

The main use is for developers of embedded applications with kweb. They can also run media without starting a terminal first (but then there are no keyboard controls available). This can be used in digital signage applications, for example, to run music in the background or to overlay video to certain areas of a web page (with especially set up presets).

There's a preset installed for the windowless method named “nogui”.

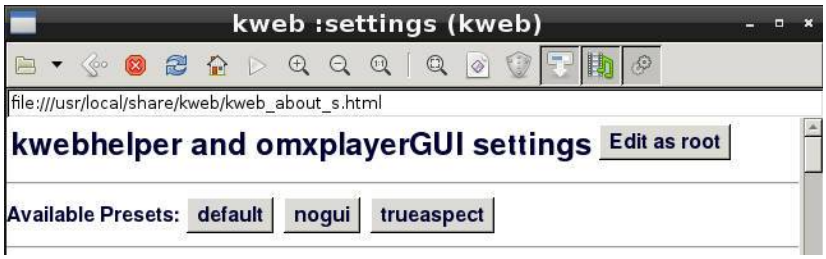
7) omxplayerGUI settings

omxplayerGUI can be fine tuned in many ways using a web interface (settings page) in Minimal Kiosk Browser (kweb). OmxplayerGUI shares its global settings with other utilities from the kweb environment. When editing the properties of omxplayerGUI you can simply ignore these additional settings.

The omxplayerGUI front end can open this settings page by calling kweb with a special configuration (as root for easier editing).

You can also call this page from inside kweb, either from its menu page, the control panel or by entering ":s" in the URL entry line. See the kweb manual for details.

a) Choosing presets



On top of the web page, you'll see a few buttons with "presets", which can be chosen with a mouse click. After selecting a preset, you have to reload the web page to see, which options have been set. Three presets are predefined: "default" (default settings), "nogui" (windowless mode) and "trueaspect" (use letterbox aspect-mode inside windows). You'll see later, how you can create your own presets.

b) List of all options used by omxplayerGUI

The following list will show all settings for omxplayerGUI in the same way as they are shown in the web interface. Each option can be edited and saved separately.

The number of settings options may be rather overwhelming for a beginner, but many of them are only of interest for a developer of embedded applications. The following options may be useful for a normal user:

Interface (all): **fontname, fontheight**
Interface (audio player): **maxlines, lwidth**
Interface (video player): **videoheight, screenmode, videomode**
Common player options: **defaultaudiovolume, autoplay, autofinish**

If you need to send special options to omxplayer, you can enter them in 4 different places:
omxoptions (video player)
omxaudiooptions (audio player)
omx_livetv_options (video player, special settings for live TV streams)
youtube_omxoptions (video player, used for web video)

GENERAL OMXPLAYER AUDIO VIDEO OPTIONS

Options for omxplayer to be used when playing video

omxoptions: Enter one element per line!

save

You can add most of the options of omxplayer, but some are filtered out in window mode.

For options requiring arguments, the argument must be entered on a separate line!

Options for omxplayer to be used when playing audio

omxaudiooptions: Enter one element per line!

save

You can add most of the options of omxplayer, but some are filtered out in window mode.

For options requiring arguments, the argument must be entered on a separate line!

Special options for watching live tv streams (omxplayer)

omx_livetv_options: Enter one element per line!

```
--live
```

save

You can add most of the options of omxplayer, but some are filtered out in window mode.

For options requiring arguments, the argument must be entered on a separate line!

Add the start of your live tv stream links to this list to enable live tv options

live_tv: Enter one element per line!

save

Example: If you have a live TV streamer running on

http://192.168.0.1:9080

add this (partial) URL to the list.

For all URLs starting with this partial URL omx_livetv_options (previous setting) will be used instead of normal omxoptions.

Mimetypes: if given, this will restrict what omxplayer will be given to play.

mimetypes: Enter one element per line!

save

Kweb sends mimetypes of all media URLs to omxplayerGUI. If you add certain mimetypes here, only those will be accepted.

Only useful for developers of embedded applications!

If omxplayerGUI is not used omxplayer is started from a terminal (xterm) to clear the screen and get full keyboard control.
Set the following to "False" to use omxplayer for video without starting a terminal first (if omxplayerGUI is not used)

Only useful for developers of embedded applications!

omxplayer_in_terminal_for_video: ◉True ○False save

Set the following to "False" to use omxplayer for audio without starting a terminal first (if omxaudioplayer is not used)

Only useful for developers of embedded applications!

omxplayer_in_terminal_for_audio: ◉True ○False save

The following list will be used, to detect audio files, especially in m3u playlists

If you are using audio file formats (extensions), which are not in this list, you can add them here.

audioextensions: Enter one element per line!

```
mp3
aac
flac
wav
wma
cda
ogg
ogm
ac3
ape
```

save

How unknown streams should be handled, must be either 'video' or 'audio'

This is also used as default setting for the video player window's "Stream:" menu.

streammode: video save

If streammode is set to "video", the following list will be used for checking for video files

If you are using video file formats (extensions), which are not in this list, you can add them here.

videoextensions: Enter one element per line!

```
asf
avi
mpg
mp4
mpeg
m2v
m1v
vob
divx
xvid
mov
m4v
m2p
```

save

If the following is set to "True" VLC will be used to play audio files and playlists (audio only)

Optionally kweb can use VLC (if installed) to play audio files and playlists containing only audio streams.

Will be ignored by the standalone player frontend!

useVLC: ○True ◉False save

---

omxplayerGUI AUDIO & VIDEO OPTIONS

If this is set to "False", kweb will use the old method of running omxplayer without a GUI for audio.

Will be ignored by the standalone player frontend!

Play audio files or playlists that contain only audio files in omxaudioplayer

useAudioplayer: ◉True ○False save

Use GUI for playing videos

If this is set to "False", kweb will use the old method of running omxplayer without a GUI for audio.

Will be ignored by the standalone player frontend!

useVideoplayer: ◉True ○False save

Volume setting when starting omxplayerGUI ranging from -20 to 4 (-60 to +12 db)

This value will be used for audio and video player windows.

defaultaudiovolume: 0 save

Start playing the first (or only) file automatically

autoplay: ◉True ○False save

Close the GUI if the last (or only) file has been played to the end

This closes a player window, not the standalone frontend.

autofinish: ◉True ○False save

Interface settings for omxaudioplayer and omxplayerGUI (video)
The font to be used for playlist and buttons

You should only use a font that is available on your system!

fontname: SansSerif save

Font size between 10 and 22, will also determine the size of the GUI window:

fontheight: 12 save

Number of entries displayed in playlist window, between 5 and 25:

This defines the visible vertical size of a playlist in an omxaudioplayer window.

maxlines: 8 save

Width of the window, value between 40 and 80, defines the minimum number of characters of the song name displayed in the songlist (usually much more are shown!), not used for video mode

This defines the visible horizontal size of a playlist in an omxaudioplayer window.

lwidth: 40 save

Minimal height of video area (also depends on fontheight!), 288 or more:

This defines the minimal vertical size of a video player window.

videoheight: 288 save

Default 'Lines:' mode, must be one of those: 'min','max', 'full'

screenmode: max save

Default video mode: set this to 'full' or 'refresh' for full screen,
to 'auto' (for automatic detection of the aspect ration) or to one of these:
'4:3','16:9','16:10','2.21:1','2.35:1','2.39:1' to play in a window
(you can also add one additional value here):

videomode: 16:9 save

Set the following to "True" for simple mode (no window resizing, moving etc. while playing video);
must be set to “True” for older omxplayer versions

Should not be used any more, but may be required if you use a very old version of omxplayer.

freeze_window: ○True ◉False save

Get aspect ratio in background if True (if videomode not one of 'auto', 'full' or 'refresh').
This costs some processing power and may even block or crash the system, especially with large AVI files.
Therefore it is disabled by default. Use it with care.

This should not be used any more!

Only remains for compatibility reasons.

get_DAR: ○True ◉False save

If the following is set to “True”, all control elements are hidden (can be enabled later on with ALT+h)

This setting only applies to video player windows.

hide_controls: ○True ◉False save

ONLINE VIDEO OPTIONS

Options for pages containing video, either HTML5 video tags or all websites supported by youtube-dl.
If html5 video tags include more than one source format, select the preferred one here.

preferred_html5_video_format: .mp4 save

Choose whether HTML5 URL extraction is tried first and youtube-dl extraction afterwards or vice versa.

HTML5 video URL extraction is faster, but only works for certain pages.

html5_first: ○True ◉False save

Additional youtube-dl options, e. g. selecting a resolution or file format

youtube_dl_options: Enter one element per line!

```
-f
best
```

save

You should only change the default setting if you really know, what you are doing, or web video might stop working.

Special omxplayer options for web video

youtube_omxoptions: Enter one element per line!

save

You can add most of the options of omxplayer, but some are filtered out in window mode.

For options requiring arguments, the argument must be entered on a separate line!

Use youtube-dl-server, if possible; also required for autostart from the frontend

use_ytdl_server: ◉True ○False save

Using the youtube-dl-server speeds up access to web video streams by about 2-3 seconds.

Requires the github version of youtube-dl!

Set this to "False", if you don't use omxplayerGUI for web video (or rarely) to spare resources.

Port on which youtube-dl-server is running.
You should only change this, if the port is used by another application.

ytdl_server_port: 9192 save

Host name or IP of youtube-dl-server.
Only change this, if you want to use one server for many clients.
If not 'localhost', this will also prevent autostart from the frontend

ytdl_server_host: localhost save

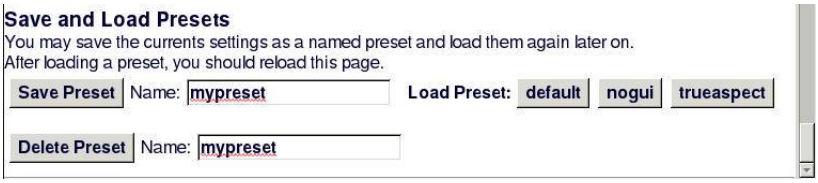
Format string to be used by youtube-dl-server. In case of missing audio, you might change this to:
best[protocol!=?m3u8][protocol!=?m3u8_native]

Only change this, if you know what you are doing!

ytdl_server_format: best save

c) Creating presets

After editing a number of options you can save the complete actual settings as a preset, which can later be loaded with a simple mouse click.



To save a preset, enter a name and click “Save Preset”. To overwrite it again later with different settings, simply use the same name. To delete a preset, enter its name and click “Delete Preset”.

To load a preset, simply click on the button with its name. Reload the page afterwards to see the active settings. This function is also available at the top of the settings page.

Note: It's highly recommended to save your own settings as named presets, because otherwise your settings will be lost when upgrading the kweb package.

Appendix A: Keyboard commands

Special omxplayerGUI Keyboard Controls

Key	Mode	Action
ALT c, q	any	quit player
ALT k	playing	kill omxplayer instance, when blocking
ALT m	any	toggle window between max and min
ALT f	any	toggle window between full and max
ALT 0	any	window min size
ALT 1...9	any	other window sizes
ALT s	playlist	save playlist
ALT u	playlist	toggle between simple (f) and extended mode
ALT h	any	hide / show controls
ALT a	any	set “Mode:” (aspect ratio) to “auto”
ALT KP+, PgUp	playlist	layer number + 1
ALT KP-, PgDown	playlist	layer number - 1
ALT +	any	next Mode: (aspect ratio)
ALT -	any	previous Mode: (aspect ratio)

omxplayerGUI Keyboard Controls (Play Mode)

OmxplayerGUI uses almost the same keyboards commands as omxplayer

omxplayer	GUI	Action
1	1	decrease speed
2	2	increase speed
<	<	rewind
>	>	fast forward
z	z	show info
j	j	previous audio stream
k	k	next audio stream
i	i	previous chapter
o	o	next chapter
n	n	previous subtitle stream
m	m	next subtitle stream
s	s	toggle subtitles
d	d	decrease subtitle delay (- 250 ms)
f	f	increase subtitle delay (+ 250 ms)
q	q, Esc	exit omxplayer
p, space	p, space, Return, Enter	pause/resume
-	-, KP-	decrease volume
+, =	+, KP+	increase volume
left arrow	left arrow	seek -30 seconds
right arrow	right arrow	seek +30 seconds
down arrow	PgDown, “,”	seek -600 seconds
up arrow	PgUp, .	seek +600 seconds
	up arrow	previous playlist item
	down arrow	next playlist item

Appendix B: Known problems

If started with hidden controls and the controls are enabled later on, switching between video aspects may not work if the window is manually resized to the smallest possible size (which is too small in this case). Select the 'min' value from the “Lines:” menu in this case.

If you toggle between hiding and showing controls in full screen or maximized (Lines:) modes, the video is not resized correctly if the controls are hidden. Hiding them before calling max or full mode works as expected.