libplayer

1.0.0

Generated by Doxygen 1.6.1

Sun Dec 6 15:28:22 2009

Contents

1 Main Page

libplayer is a multimedia A/V abstraction layer API. Its goal is to interact with Enna Media Center.

libplayer provides a generic A/V API that relies on various multimedia player for Linux systems. It currently supports `MPlayer` (through slave-mode), `xine`, `VLC` and `GStreamer`.

Its main goal is to provide an unique API that player frontends can use to control any kind of multimedia player underneath. For example, it provides a library to easily control MPlayer famous slave-mode.

1.1 MT-Level

Most functions in this API are indicated as being MT-Safe in multithreaded applications. That is right **only** if the functions are used concurrently with the same (player_t) controller. Else, unexpected behaviours can appear.

2 Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

3 File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

4 Data Structure Documentation

4.1 mrl_resource_cd_args_t Struct Reference

Arguments for audio CD.

`#include <player.h>`

4.1.1 Detailed Description

Arguments for audio CD.

Definition at line 271 of file player.h.

The documentation for this struct was generated from the following file:

- player.h

4.2 mrl_resource_local_args_t Struct Reference

Arguments for local streams.

`#include <player.h>`

4.2.1 Detailed Description

Arguments for local streams.

Definition at line 265 of file player.h.

The documentation for this struct was generated from the following file:

- player.h

4.3 mrl_resource_network_args_t Struct Reference

Arguments for network streams.

`#include <player.h>`

4.3.1 Detailed Description

Arguments for network streams.

Definition at line 308 of file player.h.

The documentation for this struct was generated from the following file:

- player.h

4.4 mrl_resource_tv_args_t Struct Reference

Arguments for radio/tv streams.

`#include <player.h>`

4.4.1 Detailed Description

Arguments for radio/tv streams.

Definition at line 295 of file player.h.

The documentation for this struct was generated from the following file:

- player.h

4.5 mrl_resource_videodisc_args_t Struct Reference

Arguments for video discs.

`#include <player.h>`

4.5.1 Detailed Description

Arguments for video discs.

Definition at line 279 of file player.h.

The documentation for this struct was generated from the following file:

- player.h

5 File Documentation

5.1 player.h File Reference

```
#include <inttypes.h>
#include <sys/types.h>
```

Data Structures

- struct mrl_resource_local_args_t
  Arguments for local streams.
- struct mrl_resource_cd_args_t
  Arguments for audio CD.
- struct mrl_resource_videodisc_args_t
  Arguments for video discs.
- struct mrl_resource_tv_args_t
  Arguments for radio/tv streams.
- struct mrl_resource_network_args_t
  Arguments for network streams.

Typedefs

- typedef struct player_s player_t
  Player controller.
- typedef struct mrl_s mrl_t
  MRL object.

Enumerations

- enum player_type_t
  Player types.

- enum player_vo_t
  Player video outputs.
- enum player_ao_t
  Player audio outputs.
- enum player_event_t
  Player events.
- enum player_verbosity_level_t
  Player verbosity.
- enum mrl_type_t
  MRL types.
- enum mrl_resource_t
  MRL resources.
- enum mrl_snapshot_t
  Snapshot image file type.
- enum mrl_metadata_type_t
  MRL metadata.
- enum mrl_metadata_cd_type_t
  MRL CDDA/CDDB metadata.
- enum mrl_metadata_dvd_type_t
  MRL DVD/DVDNAV metadata.
- enum mrl_properties_type_t
  MRL properties.
- enum player_mrl_add_t
  Player MRL add mode.
- enum player_pb_t
  Player playback mode.
- enum player_loop_t
  Player loop mode.
- enum player_framedrop_t
  Player frame dropping mode.
- enum player_x_window_flags_t
  Player X11 window flags.
- enum player_pb_state_t
  Player playback state.

- enum player_pb_seek_t
  Player playback seek mode.
- enum player_mute_t
  Player mute state.
- enum player_video_aspect_t
  Player video aspect.
- enum player_sub_alignment_t
  Player subtitle alignment.
- enum player_dvdnav_t
  Player DVDnav commands.
- enum player_vdr_t
  Player VDR commands.

Functions

Player (Un)Initialization.

- player_t * player_init (player_type_t type, player_ao_t ao, player_vo_t vo, player_verbosity_level_t verbosity, unsigned long winid, int(*event_cb)(player_event_t e, void *data))
  Initialization of a new player controller.
- void player_uninit (player_t *player)
  Uninitialization of a player controller.
- void player_set_verbosity (player_t *player, player_verbosity_level_t level)
  Set verbosity level.

Media Resource Locater (MRL) Helpers.

- mrl_t * mrl_new (player_t *player, mrl_resource_t res, void *args)
  Create a new MRL object.
- void mrl_add_subtitle (player_t *player, mrl_t *mrl, char *subtitle)
  Add a subtitle file to a MRL object.
- void mrl_free (player_t *player, mrl_t *mrl)
  Free a MRL object.
- mrl_type_t mrl_get_type (player_t *player, mrl_t *mrl)
  Get type of the stream.
- mrl_resource_t mrl_get_resource (player_t *player, mrl_t *mrl)
  Get resource of the stream.
- char * mrl_get_metadata (player_t *player, mrl_t *mrl, mrl_metadata_type_t m)

Get metadata of the stream.

- char * mrl_get_metadata_cd_track (player_t *player, mrl_t *mrl, int trackid, uint32_t *length)
  Get metadata of a track with CDDA/CDDB MRL object.
- uint32_t mrl_get_metadata_cd (player_t *player, mrl_t *mrl, mrl_metadata_cd_type_t m)
  Get metadata of a CDDA/CDDB MRL object.
- uint32_t mrl_get_metadata_dvd_title (player_t *player, mrl_t *mrl, int titleid, mrl_metadata_dvd_type_t m)
  Get metadata of a title with DVD/DVDNAV MRL object.
- char * mrl_get_metadata_dvd (player_t *player, mrl_t *mrl, uint8_t *titles)
  Get metadata of a DVD/DVDNAV MRL object.
- int mrl_get_metadata_subtitle (player_t *player, mrl_t *mrl, int pos, uint32_t *id, char **name, char **lang)
  Get subtitle metadata of the MRL object.
- uint32_t mrl_get_metadata_subtitle_nb (player_t *player, mrl_t *mrl)
  Get the number of available subtitles.
- int mrl_get_metadata_audio (player_t *player, mrl_t *mrl, int pos, uint32_t *id, char **name, char **lang)
  Get audio metadata of the MRL object.
- uint32_t mrl_get_metadata_audio_nb (player_t *player, mrl_t *mrl)
  Get the number of available audio streams.
- uint32_t mrl_get_property (player_t *player, mrl_t *mrl, mrl_properties_type_t p)
  Get property of the stream.
- char * mrl_get_audio_codec (player_t *player, mrl_t *mrl)
  Get audio codec name of the stream.
- char * mrl_get_video_codec (player_t *player, mrl_t *mrl)
  Get video codec name of the stream.
- off_t mrl_get_size (player_t *player, mrl_t *mrl)
  Get size of the resource.
- void mrl_video_snapshot (player_t *player, mrl_t *mrl, int pos, mrl_snapshot_t t, const char *dst)
  Take a video snapshot.

Player to MRL connection.

- mrl_t * player_mrl_get_current (player_t *player)
  Get current MRL set in the internal playlist.
- void player_mrl_set (player_t *player, mrl_t *mrl)
  Set MRL object in the internal playlist.
- void player_mrl_append (player_t *player, mrl_t *mrl, player_mrl_add_t when)

Append MRL object in the internal playlist.

- void player_mrl_remove (player_t *player)
  Remove current MRL object in the internal playlist.
- void player_mrl_remove_all (player_t *player)
  Remove all MRL objects in the internal playlist.
- void player_mrl_previous (player_t *player)
  Go the the previous MRL object in the internal playlist.
- void player_mrl_next (player_t *player)
  Go the the next MRL object in the internal playlist.

Player tuning & properties.

- int player_get_time_pos (player_t *player)
  Get current time position in the current stream.
- int player_get_percent_pos (player_t *player)
  Get percent position in the current stream.
- void player_set_playback (player_t *player, player_pb_t pb)
  Set playback mode.
- void player_set_loop (player_t *player, player_loop_t loop, int value)
  Set loop mode and value.
- void player_set_shuffle (player_t *player, int value)
  Shuffle playback in the internal playlist.
- void player_set_framedrop (player_t *player, player_framedrop_t fd)
  Set frame dropping with video playback.
- void player_set_mouse_position (player_t *player, int x, int y)
  Set the mouse position to the player.
- void player_x_window_set_properties (player_t *player, int x, int y, int w, int h, int flags)
  Set properties of X11 window handled by libplayer.
- void player_osd_show_text (player_t *player, const char *text, int x, int y, int duration)
  Show a text on the On-screen Display.

Playback related controls.

- player_pb_state_t player_playback_get_state (player_t *player)
  Get current playback state.
- void player_playback_start (player_t *player)
  Start a new playback.
- void player_playback_stop (player_t *player)
  Stop playback.

- void player_playback_pause (player_t *player)
  Pause and unpause playback.
- void player_playback_seek (player_t *player, int value, player_pb_seek_t seek)
  Seek in the stream.
- void player_playback_seek_chapter (player_t *player, int value, int absolute)
  Seek chapter in the stream.
- void player_playback_speed (player_t *player, float value)
  Change playback speed.

Audio related controls.

- int player_audio_volume_get (player_t *player)
  Get current volume.
- void player_audio_volume_set (player_t *player, int value)
  Set volume.
- player_mute_t player_audio_mute_get (player_t *player)
  Get mute state.
- void player_audio_mute_set (player_t *player, player_mute_t value)
  Set mute state.
- void player_audio_set_delay (player_t *player, int value, int absolute)
  Set audio delay.
- void player_audio_select (player_t *player, int audio_id)
  Select audio ID.
- void player_audio_prev (player_t *player)
  Select the previous audio ID.
- void player_audio_next (player_t *player)
  Select the next audio ID.

Video related controls.

- void player_video_set_fullscreen (player_t *player, int value)
  Set video in fullscreen.
- void player_video_set_aspect (player_t *player, player_video_aspect_t aspect, int8_t value, int absolute)
  Set video aspect.
- void player_video_set_panscan (player_t *player, int8_t value, int absolute)
  Set video panscan.
- void player_video_set_aspect_ratio (player_t *player, float value)
  Set video aspect ratio.

Subtitles related controls.

- void player_subtitle_set_delay (player_t *player, int value)
  Set subtitle delay.
- void player_subtitle_set_alignment (player_t *player, player_sub_alignment_t a)
  Set subtitle alignment.
- void player_subtitle_set_position (player_t *player, int value)
  Set subtitle position.
- void player_subtitle_set_visibility (player_t *player, int value)
  Set subtitle visibility.
- void player_subtitle_scale (player_t *player, int value, int absolute)
  Set subtitle scale.
- void player_subtitle_select (player_t *player, int sub_id)
  Select subtitle ID.
- void player_subtitle_prev (player_t *player)
  Select the previous subtitle ID.
- void player_subtitle_next (player_t *player)
  Select the next subtitle ID.

DVD specific controls.

- void player_dvd_nav (player_t *player, player_dvdnav_t value)
  DVD Navigation commands.
- void player_dvd_angle_select (player_t *player, int angle)
  Select DVD angle.
- void player_dvd_angle_prev (player_t *player)
  Select the previous DVD angle.
- void player_dvd_angle_next (player_t *player)
  Select the next DVD angle.
- void player_dvd_title_select (player_t *player, int title)
  Select DVD title.
- void player_dvd_title_prev (player_t *player)
  Select the previous DVD title.
- void player_dvd_title_next (player_t *player)
  Select the next DVD title.

TV/DVB specific controls.

- void player_tv_channel_select (player_t *player, const char *channel)
  Select TV channel.
- void player_tv_channel_prev (player_t *player)
  Select the previous TV channel.
- void player_tv_channel_next (player_t *player)
  Select the next TV channel.

Radio specific controls.

- void player_radio_channel_select (player_t *player, const char *channel)
  Select radio channel.
- void player_radio_channel_prev (player_t *player)
  Select the previous radio channel.
- void player_radio_channel_next (player_t *player)
  Select the next radio channel.

VDR specific controls.

- void player_vdr (player_t *player, player_vdr_t value)
  VDR commands.

Global libplayer functions.

- int libplayer_wrapper_enabled (player_type_t type)
  Test if a wrapper is enabled.
- int libplayer_wrapper_supported_res (player_type_t type, mrl_resource_t res)
  Test if a resource is supported by a wrapper.

5.1.1 Detailed Description

GeeXboX libplayer public API header.

Definition in file player.h.

5.1.2 Typedef Documentation

5.1.2.1 typedef struct mrl_s mrl_t

MRL object. This handles an audio, video or image resource.

Definition at line 210 of file player.h.

5.1.2.2 typedef struct player_s player_t

Player controller. This controls a multimedia player.
Definition at line 90 of file player.h.

5.1.3 Enumeration Type Documentation

5.1.3.1 enum mrl_metadata_cd_type_t

MRL CDDA/CDDB metadata.
Definition at line 335 of file player.h.

5.1.3.2 enum mrl_metadata_dvd_type_t

MRL DVD/DVDNAV metadata.
Definition at line 341 of file player.h.

5.1.3.3 enum mrl_metadata_type_t

MRL metadata.
Definition at line 324 of file player.h.

5.1.3.4 enum mrl_properties_type_t

MRL properties.
Definition at line 348 of file player.h.

5.1.3.5 enum mrl_resource_t

MRL resources.
Definition at line 227 of file player.h.

5.1.3.6 enum mrl_snapshot_t

Snapshot image file type.
Definition at line 316 of file player.h.

5.1.3.7 enum mrl_type_t

MRL types.
Definition at line 213 of file player.h.

5.1.3.8 enum player_ao_t

Player audio outputs.
Definition at line 114 of file player.h.

5.1.3.9 enum player_dvdnav_t

Player DVDnav commands.
Definition at line 1386 of file player.h.

5.1.3.10 enum player_event_t

Player events.
Definition at line 122 of file player.h.

5.1.3.11 enum player_framedrop_t

Player frame dropping mode.
Definition at line 787 of file player.h.

5.1.3.12 enum player_loop_t

Player loop mode.
Definition at line 780 of file player.h.

5.1.3.13 enum player_mrl_add_t

Player MRL add mode.
Definition at line 682 of file player.h.

5.1.3.14 enum player_mute_t

Player mute state.

Definition at line 1059 of file player.h.

5.1.3.15 enum player_pb_seek_t

Player playback seek mode.

Definition at line 953 of file player.h.

5.1.3.16 enum player_pb_state_t

Player playback state.

Definition at line 946 of file player.h.

5.1.3.17 enum player_pb_t

Player playback mode.

Definition at line 774 of file player.h.

5.1.3.18 enum player_sub_alignment_t

Player subtitle alignment.

Definition at line 1261 of file player.h.

5.1.3.19 enum player_type_t

Player types.

Definition at line 93 of file player.h.

5.1.3.20 enum player_vdr_t

Player VDR commands.

Definition at line 1607 of file player.h.

5.1.3.21 enum player_verbosity_level_t

Player verbosity.

Definition at line 133 of file player.h.

5.1.3.22 enum player_video_aspect_t

Player video aspect.

Definition at line 1182 of file player.h.

5.1.3.23 enum player_vo_t

Player video outputs.

Definition at line 102 of file player.h.

5.1.3.24 enum player_x_window_flags_t

Player X11 window flags.

Definition at line 794 of file player.h.

5.1.4 Function Documentation

5.1.4.1 int libplayer_wrapper_enabled (player_type_t *type*)

Test if a wrapper is enabled.

Warning:

MT-Safe in multithreaded applications.

Parameters:

← ***type*** Player type.

Returns:

1 if enabled, 0 otherwise.

5.1.4.2 int libplayer_wrapper_supported_res (player_type_t *type*, mrl_resource_t *res*)

Test if a resource is supported by a wrapper.

Warning:

MT-Safe in multithreaded applications.

Parameters:

← ***type*** Player type.

← ***res*** Resource type.

Returns:

1 if supported, 0 otherwise.

5.1.4.3 void mrl_add_subtitle (player_t * *player*, mrl_t * *mrl*, char * *subtitle*)

Add a subtitle file to a MRL object.

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***mrl*** MRL object, NULL for current.

← ***subtitle*** Location of the subtitle file to be added.

5.1.4.4 void mrl_free (player_t * *player*, mrl_t * *mrl*)

Free a MRL object. Never use this function when the MRL (or a linked MRL) is set in the playlist of a player controller.

Warning:

Must be used only as the last mrl function for one MRL object.
MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***mrl*** MRL object.

5.1.4.5 char* mrl_get_audio_codec (player_t * *player*, mrl_t * *mrl*)

Get audio codec name of the stream. Wrappers supported (even partially): MPlayer, xine

Warning:

The returned pointer must be freed when no longer used.
MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***mrl*** MRL object, NULL for current.

Returns:

Audio codec name, NULL otherwise.

5.1.4.6 char* mrl_get_metadata (player_t * *player*, mrl_t * *mrl*, mrl_metadata_type_t *m*)

Get metadata of the stream. This function can be slow when the stream is not (fastly) reachable.

Wrappers supported (even partially): MPlayer, VLC, xine

Warning:

The returned pointer must be freed when no longer used.
MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***mrl*** MRL object, NULL for current.

← ***m*** Type of metadata to get.

Returns:

Metadata string, NULL otherwise.

5.1.4.7 int mrl_get_metadata_audio (player_t * *player*, mrl_t * *mrl*, int *pos*, uint32_t * *id*, char ** *name*, char ** *lang*)

Get audio metadata of the MRL object. This function can be slow when the stream is not (fastly) reachable.

The `pos` argument is the position of the audio stream in the internal list of libplayer. The first audio stream begins with 1. `id` returned by this function can be used with player_audio_select().

Wrappers supported (even partially): MPlayer

Warning:

The pointers (`name` and `lang`) must be freed when no longer used.
MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***mrl*** MRL object, NULL for current.
← ***pos*** Position of the audio stream.
→ ***id*** ID of the audio stream, NULL to ignore.
→ ***name*** Name of the audio stream, NULL to ignore.
→ ***lang*** Language of the audio stream, NULL to ignore.

Returns:

1 for success, 0 if the audio stream is not available.

5.1.4.8 uint32_t mrl_get_metadata_audio_nb (player_t * *player*, mrl_t * *mrl*)

Get the number of available audio streams. This function can be slow when the stream is not (fastly) reachable.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***mrl*** MRL object, NULL for current.

Returns:

Number of audio streams.

5.1.4.9 uint32_t mrl_get_metadata_cd (player_t * *player*, mrl_t * *mrl*, mrl_metadata_cd_type_t *m*)

Get metadata of a CDDA/CDDB MRL object. This function can be slow when the stream is not (fastly) reachable.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***mrl*** MRL object, NULL for current.
← ***m*** Type of metadata to get.

Returns:

Metadata value.

5.1.4.10 char* mrl_get_metadata_cd_track (player_t * *player*, mrl_t * *mrl*, int *trackid*, uint32_t * *length*)

Get metadata of a track with CDDA/CDDB MRL object. This function can be slow when the stream is not (fastly) reachable.

Wrappers supported (even partially): MPlayer

Warning:

The returned pointer must be freed when no longer used.
MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***mrl*** MRL object, NULL for current.

← ***trackid*** Track ID on the CD.

→ ***length*** Length of the track (millisecond).

Returns:

Title of the track (CDDB only), NULL otherwise.

5.1.4.11 char* mrl_get_metadata_dvd (player_t * *player*, mrl_t * *mrl*, uint8_t * *titles*)

Get metadata of a DVD/DVDNAV MRL object. This function can be slow when the stream is not (fastly) reachable.

Wrappers supported (even partially): MPlayer, xine

Warning:

The returned pointer must be freed when no longer used.
MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***mrl*** MRL object, NULL for current.

→ ***titles*** How many titles on the DVD.

Returns:

Volume ID, NULL otherwise.

5.1.4.12 uint32_t mrl_get_metadata_dvd_title (player_t * *player*, mrl_t * *mrl*, int *titleid*, mrl_metadata_dvd_type_t *m*)

Get metadata of a title with DVD/DVDNAV MRL object. This function can be slow when the stream is not (fastly) reachable.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***mrl*** MRL object, NULL for current.

← ***titleid*** Title ID on the DVD.

← ***m*** Type of metadata to get.

Returns:

Metadata value.

5.1.4.13 int mrl_get_metadata_subtitle (player_t * *player*, mrl_t * *mrl*, int *pos*, uint32_t * *id*, char ** *name*, char ** *lang*)

Get subtitle metadata of the MRL object. This function can be slow when the stream is not (fastly) reachable.

The `pos` argument is the position of the subtitle in the internal list of libplayer. The first subtitle begins with 1. `id` returned by this function can be used with player_subtitle_select().

Wrappers supported (even partially): MPlayer

Warning:

The pointers (`name` and `lang`) must be freed when no longer used.
MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***mrl*** MRL object, NULL for current.

← ***pos*** Position of the subtitle.

→ ***id*** ID of the subtitle, NULL to ignore.

→ ***name*** Name of the subtitle, NULL to ignore.

→ ***lang*** Language of the subtitle, NULL to ignore.

Returns:

1 for success, 0 if the subtitle is not available.

5.1.4.14 uint32_t mrl_get_metadata_subtitle_nb (player_t * *player*, mrl_t * *mrl*)

Get the number of available subtitles. This function can be slow when the stream is not (fastly) reachable.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***mrl*** MRL object, NULL for current.

Returns:

Number of subtitles.

5.1.4.15 uint32_t mrl_get_property (player_t * *player*, mrl_t * *mrl*, mrl_properties_type_t *p*)

Get property of the stream. Wrappers supported (even partially): MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***mrl*** MRL object, NULL for current.
← ***p*** Type of property.

Returns:

Property value.

5.1.4.16 mrl_resource_t mrl_get_resource (player_t * *player*, mrl_t * *mrl*)

Get resource of the stream.

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***mrl*** MRL object, NULL for current.

Returns:

Resource of MRL object.

5.1.4.17 off_t mrl_get_size (player_t * *player*, mrl_t * *mrl*)

Get size of the resource. Wrappers supported (even partially): MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***mrl*** MRL object, NULL for current.

Returns:

Size of the stream (bytes).

5.1.4.18 mrl_type_t mrl_get_type (player_t * *player*, mrl_t * *mrl*)

Get type of the stream.

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***mrl*** MRL object, NULL for current.

Returns:

Type of MRL object.

5.1.4.19 char* mrl_get_video_codec (player_t * *player*, mrl_t * *mrl*)

Get video codec name of the stream. Wrappers supported (even partially): MPlayer, xine

Warning:

The returned pointer must be freed when no longer used.
MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***mrl*** MRL object, NULL for current.

Returns:

Video codec name, NULL otherwise.

5.1.4.20 mrl_t* mrl_new (player_t * *player*, mrl_resource_t *res*, void * *args*)

Create a new MRL object. This function can be slow when the stream is not (fastly) reachable.

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***res*** Resource type.
← ***args*** Arguments specific to the resource type.

Returns:

MRL object, NULL otherwise.

5.1.4.21 void mrl_video_snapshot (player_t * *player*, mrl_t * *mrl*, int *pos*, mrl_snapshot_t *t*, const char * *dst*)

Take a video snapshot. One frame at the `pos` (in second) is saved to `dst`.

Wrappers supported (even partially): MPlayer, VLC

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***mrl*** MRL object, NULL for current.
← ***pos*** Time position (second).
← ***t*** Image file type.
← ***dst*** Destination file, NULL for default filename in the current directory.

5.1.4.22 player_mute_t player_audio_mute_get (player_t * *player*)

Get mute state. Wrappers supported (even partially): MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

Returns:

Mute state.

5.1.4.23 void player_audio_mute_set (player_t * *player*, player_mute_t *value*)

Set mute state. Wrappers supported (even partially): MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***value*** Mute state to set.

5.1.4.24 void player_audio_next (player_t * *player*)

Select the next audio ID. It stays on the same audio ID if no next stream exists.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.25 void player_audio_prev (player_t * *player*)

Select the previous audio ID. It stays on the same audio ID if no previous stream exists.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.26 void player_audio_select (player_t * *player*, int *audio_id*)

Select audio ID. Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***audio_id*** ID of the audio stream to select.

5.1.4.27 void player_audio_set_delay (player_t * *player*, int *value*, int *absolute*)

Set audio delay. Only useful with video files to set delay between audio and video streams.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***value*** Delay to set (millisecond).

← ***absolute*** Mode, 0 for relative.

5.1.4.28 int player_audio_volume_get (player_t * *player*)

Get current volume. Wrappers supported (even partially): MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

Returns:

Volume (percent).

5.1.4.29 void player_audio_volume_set (player_t * *player*, int *value*)

Set volume. Wrappers supported (even partially): MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***value*** Volume to set (percent).

5.1.4.30 void player_dvd_angle_next (player_t * *player*)

Select the next DVD angle. It stays on the same if no next angle exists.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.31 void player_dvd_angle_prev (player_t * *player*)

Select the previous DVD angle. It stays on the same if no previous angle exists.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.32 void player_dvd_angle_select (player_t * *player*, int *angle*)

Select DVD angle. Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***angle*** Angle to select.

5.1.4.33 void player_dvd_nav (player_t * *player*, player_dvdnav_t *value*)

DVD Navigation commands. Wrappers supported (even partially): MPlayer, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***value*** Command to send.

5.1.4.34 void player_dvd_title_next (player_t * *player*)

Select the next DVD title. It stays on the same if no next title exists.

Wrappers supported (even partially): VLC

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.35 void player_dvd_title_prev (player_t * *player*)

Select the previous DVD title. It stays on the same if no previous title exists.

Wrappers supported (even partially): VLC

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.36 void player_dvd_title_select (player_t * *player*, int *title*)

Select DVD title. Wrappers supported (even partially): MPlayer, VLC

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***title*** Title to select.

5.1.4.37 int player_get_percent_pos (player_t * *player*)

Get percent position in the current stream. Wrapper supported (even partially): MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

Returns:

Percent position.

5.1.4.38 int player_get_time_pos (player_t * *player*)

Get current time position in the current stream. Wrappers supported (even partially): MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

Returns:

Time position (millisecond).

5.1.4.39 player_t* player_init (player_type_t *type*, player_ao_t *ao*, player_vo_t *vo*, player_verbosity_level_t *verbosity*, unsigned long *winid*, int(*)(player_event_t e, void *data) *event_cb*)

Initialization of a new player controller. Multiple player controllers can be initialized with any wrappers. The same Window ID can be used to attach their video.

Wrappers supported (even partially): GStreamer, MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***type*** Type of wrapper to load.

← ***ao*** Audio output driver to use.

← ***vo*** Video output driver to use.

← ***verbosity*** Level of verbosity to set.

← ***winid*** WinID to attach the video (X Window), 0 to disable.

← ***event_cb*** Public callback, NULL to disable.

Returns:

Player controller, NULL otherwise.

5.1.4.40 void player_mrl_append (player_t * *player*, mrl_t * *mrl*, player_mrl_add_t *when*)

Append MRL object in the internal playlist.

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***mrl*** MRL object to append.

← ***when*** Just append, or append and go to the end to play.

5.1.4.41 mrl_t* player_mrl_get_current (player_t * *player*)

Get current MRL set in the internal playlist.

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

Returns:

MRL object.

5.1.4.42 void player_mrl_next (player_t * *player*)

Go the the next MRL object in the internal playlist. Playback is started if a next MRL object exists.

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.43 void player_mrl_previous (player_t * *player*)

Go the the previous MRL object in the internal playlist. Playback is started if a previous MRL object exists.

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.44 void player_mrl_remove (player_t * *player*)

Remove current MRL object in the internal playlist. Current MRL object is freed on the way.

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.45 void player_mrl_remove_all (player_t * *player*)

Remove all MRL objects in the internal playlist. All MRL objects are freed on the way.

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.46 void player_mrl_set (player_t * *player*, mrl_t * *mrl*)

Set MRL object in the internal playlist. If a MRL was already set in the playlist, then the current is freed and replaced by the new MRL object.

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***mrl*** MRL object to set.

5.1.4.47 void player_osd_show_text (player_t * *player*, const char * *text*, int *x*, int *y*, int *duration*)

Show a text on the On-screen Display. Coordinates are not usable with MPlayer wrapper. The text is always shown from the top-left corner.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***text*** Text to show on the OSD.

← ***x*** X coordinate (pixel).

← ***y*** Y coordinate (pixel).

← ***duration*** Duration (millisecond).

5.1.4.48 player_pb_state_t player_playback_get_state (player_t * *player*)

Get current playback state.

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

Returns:

Playback state.

5.1.4.49 void player_playback_pause (player_t * *player*)

Pause and unpause playback. Wrappers supported (even partially): MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.50 void player_playback_seek (player_t * *player*, int *value*, player_pb_seek_t *seek*)

Seek in the stream. Wrappers supported (even partially): MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***value*** Value for seeking (second or percent).
← ***seek*** Seeking mode.

5.1.4.51 void player_playback_seek_chapter (player_t * *player*, int *value*, int *absolute*)

Seek chapter in the stream. Wrappers supported (even partially): MPlayer, VLC

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***value*** Value for seeking.
← ***absolute*** Mode, 0 for relative.

5.1.4.52 void player_playback_speed (player_t * *player*, float *value*)

Change playback speed. This function can't be used to play in backward.

Wrappers supported (even partially): MPlayer, VLC

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***value*** Factor of playback speed to set.

5.1.4.53 void player_playback_start (player_t * *player*)

Start a new playback. The playback is always started from the beginning.

Wrappers supported (even partially): GStreamer, MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.54 void player_playback_stop (player_t * *player*)

Stop playback. Wrappers supported (even partially): GStreamer, MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.55 void player_radio_channel_next (player_t * *player*)

Select the next radio channel. It stays on the same if no next channel exists.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.56 void player_radio_channel_prev (player_t * *player*)

Select the previous radio channel. It stays on the same if no previous channel exists.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.57 void player_radio_channel_select (player_t * *player*, const char * *channel*)

Select radio channel. Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***channel*** Channel to select.

5.1.4.58 void player_set_framedrop (player_t * *player*, player_framedrop_t *fd*)

Set frame dropping with video playback. Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***fd*** Frame dropping type to set.

5.1.4.59 void player_set_loop (player_t * *player*, player_loop_t *loop*, int *value*)

Set loop mode and value. Only enabled if playback mode is auto, see player_set_playback().

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***loop*** Mode to use (one element or the whole playlist).
← ***value*** How many loops, negative for infinite.

5.1.4.60 void player_set_mouse_position (player_t * *player*, int *x*, int *y*)

Set the mouse position to the player. The main goal is to select buttons in DVD menu. The coordinates are relative to the top-left corner of the root window. The root window is `winid` passed with player_init().

Wrappers supported (even partially): MPlayer, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***x*** X coordinate (pixel).
← ***y*** Y coordinate (pixel).

5.1.4.61 void player_set_playback (player_t * *player*, player_pb_t *pb*)

Set playback mode. If the playback mode is set to PLAYER_PB_AUTO, then loop and shuffle can be used with the internal playlist. By default, AUTO will just going to the next available MRL object in the playlist and start a new playback.

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***pb*** Mode to use.

5.1.4.62 void player_set_shuffle (player_t * *player*, int *value*)

Shuffle playback in the internal playlist. Only enabled if playback mode is auto, see player_set_playback().

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***value*** Different of 0 to enable.

5.1.4.63 void player_set_verbosity (player_t * *player*, player_verbosity_level_t *level*)

Set verbosity level. Wrappers supported (even partially): MPlayer, VLC, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***level*** Level of verbosity to set.

5.1.4.64 void player_subtitle_next (player_t * *player*)

Select the next subtitle ID. It stays on the same subtitle ID if no next subtitle exists.

Wrappers supported (even partially): MPlayer, VLC

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.65 void player_subtitle_prev (player_t * *player*)

Select the previous subtitle ID. It stays on the same subtitle ID if no previous subtitle exists.

Wrappers supported (even partially): MPlayer, VLC

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.66 void player_subtitle_scale (player_t * *player*, int *value*, int *absolute*)

Set subtitle scale. Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***value*** Scale to set.
← ***absolute*** Mode, 0 for relative.

5.1.4.67 void player_subtitle_select (player_t * *player*, int *sub_id*)

Select subtitle ID. Wrappers supported (even partially): MPlayer, VLC

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***sub_id*** ID of the subtitle to select.

5.1.4.68 void player_subtitle_set_alignment (player_t * *player*, player_sub_alignment_t *a*)

Set subtitle alignment. Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***a*** Alignment to set.

5.1.4.69 void player_subtitle_set_delay (player_t * *player*, int *value*)

Set subtitle delay. Only useful with video files to set delay between audio stream and the subtitles.

Wrappers supported (even partially): MPlayer, xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.
← ***value*** Delay to set (millisecond).

5.1.4.70 void player_subtitle_set_position (player_t * *player*, int *value*)

Set subtitle position. Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***value*** Position to set.

5.1.4.71 void player_subtitle_set_visibility (player_t * *player*, int *value*)

Set subtitle visibility. Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***value*** Different of 0 to view the subtitles.

5.1.4.72 void player_tv_channel_next (player_t * *player*)

Select the next TV channel. It stays on the same if no next channel exists.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.73 void player_tv_channel_prev (player_t * *player*)

Select the previous TV channel. It stays on the same if no previous channel exists.

Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

5.1.4.74 void player_tv_channel_select (player_t * *player*, const char * *channel*)

Select TV channel. Wrappers supported (even partially): MPlayer

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***channel*** Channel to select.

5.1.4.75 void player_uninit (player_t * *player*)

Uninitialization of a player controller. All MRL objects in the internal playlist will be freed.

Wrappers supported (even partially): GStreamer, MPlayer, VLC, xine

Warning:

Must be used only as the last player function for a controller.

Parameters:

← ***player*** Player controller.

5.1.4.76 void player_vdr (player_t * *player*, player_vdr_t *value*)

VDR commands. Wrappers supported (even partially): xine

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***value*** Command to send.

5.1.4.77 void player_video_set_aspect (player_t * *player*, player_video_aspect_t *aspect*, int8_t *value*, int *absolute*)

Set video aspect. Wrappers supported (even partially): none

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***aspect*** Aspect to change.

← ***value*** Value for aspect to set.

← ***absolute*** Mode, 0 for relative.

5.1.4.78 void player_video_set_aspect_ratio (player_t * *player*, float *value*)

Set video aspect ratio. Wrappers supported (even partially): MPlayer, VLC

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***value*** Ratio to set.

5.1.4.79 void player_video_set_fullscreen (player_t * *player*, int *value*)

Set video in fullscreen. By default the video is always in fullscreen. To work with a window, use winid parameter in player_init().

Wrappers supported (even partially): MPlayer, VLC

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***value*** Different of 0 to set fullscreen.

5.1.4.80 void player_video_set_panscan (player_t * *player*, int8_t *value*, int *absolute*)

Set video panscan. Wrappers supported (even partially): none

Warning:

MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***value*** Value for panscan to set.

← ***absolute*** Mode, 0 for relative.

5.1.4.81 void player_x_window_set_properties (player_t * *player*, int *x*, int *y*, int *w*, int *h*, int *flags*)

Set properties of X11 window handled by libplayer. Origin to the top-left corner.

Wrappers supported (even partially): MPlayer, xine

Warning:

Only usable with video outputs X11 compliant.
MT-Safe in multithreaded applications (see MT-Level).

Parameters:

← ***player*** Player controller.

← ***x*** X coordinate (pixel).

← ***y*** Y coordinate (pixel).

← ***w*** Width (pixel).

← ***h*** Height (pixel).

← ***flags*** Flags to select properties to change.

Index