

playlists(3)

libmtp

playlists(3)

NAME

libmtp –

SYNOPSIS**Functions**

```

LIBMTP_playlist_t * LIBMTP_new_playlist_t (void)
void LIBMTP_destroy_playlist_t (LIBMTP_playlist_t *)
LIBMTP_playlist_t * LIBMTP_Get_Playlist_List (LIBMTP_mtpdevice_t *)
LIBMTP_playlist_t * LIBMTP_Get_Playlist (LIBMTP_mtpdevice_t *, uint32_t const)
int LIBMTP_Create_New_Playlist (LIBMTP_mtpdevice_t *, LIBMTP_playlist_t * const)
int LIBMTP_Update_Playlist (LIBMTP_mtpdevice_t *, LIBMTP_playlist_t * const)
int LIBMTP_Set_Playlist_Name (LIBMTP_mtpdevice_t *, LIBMTP_playlist_t *, const char *)

```

Detailed Description**Function Documentation**

int LIBMTP_Create_New_Playlist (**LIBMTP_mtpdevice_t *** **device**, **LIBMTP_playlist_t *** **const** **metadata**)

This routine creates a new playlist based on the metadata supplied. If the `tracks` field of the metadata contains a track listing, these tracks will be added to the playlist.

Parameters:

device a pointer to the device to create the new playlist on.
metadata the metadata for the new playlist. If the function exits with success, the `playlist_id` field of this struct will contain the new playlist ID of the playlist.

- `metadata->parent_id` should be set to the parent (e.g. folder) to store this track in. Since some devices are a bit picky about where files are placed, a default folder will be chosen if libmtp has detected one for the current filetype and this parameter is set to 0. If this is 0 and no default folder can be found, the file will be stored in the root folder.
- `metadata->storage_id` should be set to the desired storage (e.g. memory card or whatever your device presents) to store this track in. Setting this to 0 will store the track on the primary storage.

Returns:

0 on success, any other value means failure.

See also:

LIBMTP_Update_Playlist()

LIBMTP_Delete_Object()

References `LIBMTP_mtpdevice_struct::default_music_folder`, `LIBMTP_mtpdevice_struct::default_playlist_folder`, `LIBMTP_playlist_struct::parent_id`, and `LIBMTP_mtpdevice_struct::usbinfo`.

void LIBMTP_destroy_playlist_t (**LIBMTP_playlist_t *** **playlist**)

This destroys a playlist metadata structure and deallocates the memory used by it, including any strings. Never use a track metadata structure again after calling this function on it.

Parameters:

playlist the playlist metadata to destroy.

See also:

LIBMTP_new_playlist_t()

References `LIBMTP_playlist_struct::name`, and `LIBMTP_playlist_struct::tracks`.

LIBMTP_playlist_t* LIBMTP_Get_Playlist (**LIBMTP_mtpdevice_t *** **device**, **uint32_t** **const** **plid**)

This function retrieves an individual playlist from the device.

Parameters:

device a pointer to the device to get the playlist from.
plid the unique ID of the playlist to retrieve.

Returns:

a valid playlist metadata post or NULL on failure.

See also:

LIBMTP_Get_Playlist_List()

References `LIBMTP_mtpdevice_struct::usbinfo`.



playlists(3)

libmtp

playlists(3)

LIBMTP_playlist_t* LIBMTP_Get_Playlist_List (LIBMTP_mtpdevice_t * device)

This function returns a list of the playlists available on the device. Typical usage:

Parameters:

device a pointer to the device to get the playlist listing from.

Returns:

a playlist list on success, else NULL. If there are no playlists on the device, NULL will be returned as well.

See also:**LIBMTP_Get_Playlist()**

References LIBMTP_mtpdevice_struct::usbinfo.

LIBMTP_playlist_t* LIBMTP_new_playlist_t (void)

This creates a new playlist metadata structure and allocates memory for it. Notice that if you add strings to this structure they will be freed by the corresponding LIBMTP_destroy_playlist_t operation later, so be careful of using strdup() when assigning strings, e.g.:

```
LIBMTP_playlist_t *pl = LIBMTP_new_playlist_t();
```

```
pl->name = strdup(str);
```

```
....
```

```
LIBMTP_destroy_playlist_t(pl);
```

Returns:

a pointer to the newly allocated metadata structure.

See also:**LIBMTP_destroy_playlist_t()**

References LIBMTP_playlist_struct::playlist_id.

int LIBMTP_Set_Playlist_Name (LIBMTP_mtpdevice_t * device, LIBMTP_playlist_t * playlist, const char * newname)

This function renames a single playlist object file holder. This simply means that the PTP_OPC_ObjectFileName property is updated, if this is supported by the device. The playlist filename should nominally end with an extension like '.pla'.

NOTE: if you want to change the metadata the device display about a playlist you must *not* use this function, use **LIBMTP_Update_Playlist()** instead!

Parameters:

device a pointer to the device that contains the file.

playlist the playlist metadata of the playlist to rename. On success, the name member is updated. Be aware, that this name can be different than newname depending of device restrictions.

newname the new name for this object.

Returns:

0 on success, any other value means failure.

See also:**LIBMTP_Update_Playlist()****int LIBMTP_Update_Playlist (LIBMTP_mtpdevice_t * device, LIBMTP_playlist_t *const metadata)**

This routine updates a playlist based on the metadata supplied. If the `tracks` field of the metadata contains a track listing, these tracks will be added to the playlist in place of those already present, i.e. the previous track listing will be deleted. For Samsung devices the playlist id (metadata->playlist_id) is likely to change.

Parameters:

device a pointer to the device to create the new playlist on.

metadata the metadata for the playlist to be updated. notice that the field `playlist_id` must contain the appropriate playlist ID. Playlist ID be modified to a new playlist ID by the time the function returns since edit-in-place is not always possible.

Returns:

0 on success, any other value means failure.

See also:**LIBMTP_Create_New_Playlist()****LIBMTP_Delete_Object()**

References LIBMTP_mtpdevice_struct::usbinfo.

Author

Generated automatically by Doxygen for libmtp from the source code.

