

folders(3)

libmtp

folders(3)

**NAME**

libmtp –

**SYNOPSIS****Functions**

```

LIBMTP_folder_t * LIBMTP_new_folder_t (void)
void LIBMTP_destroy_folder_t (LIBMTP_folder_t *)
LIBMTP_folder_t * LIBMTP_Get_Folder_List (LIBMTP_mtpdevice_t *)
LIBMTP_folder_t * LIBMTP_Get_Folder_List_For_Storage (LIBMTP_mtpdevice_t *, uint32_t
    const)
LIBMTP_folder_t * LIBMTP_Find_Folder (LIBMTP_folder_t *, uint32_t const)
uint32_t LIBMTP_Create_Folder (LIBMTP_mtpdevice_t *, char *, uint32_t, uint32_t)
int LIBMTP_Set_Folder_Name (LIBMTP_mtpdevice_t *, LIBMTP_folder_t *, const char *)

```

**Detailed Description****Function Documentation**

**uint32\_t LIBMTP\_Create\_Folder** (**LIBMTP\_mtpdevice\_t \*** **device**, **char \*** **name**, **uint32\_t** **parent\_id**, **uint32\_t** **storage\_id**)

This create a folder on the current MTP device. The PTP name for a folder is 'association'. The PTP/MTP devices does not have an internal 'folder' concept really, it contains a flat list of all files and some file are 'associations' that other files and folders may refer to as its 'parent'.

**Parameters:**

*device* a pointer to the device to create the folder on.  
*name* the name of the new folder. Note this can be modified if the device does not support all the characters in the name.  
*parent\_id* id of parent folder to add the new folder to, or 0xFFFFFFFF to put it in the root directory.  
*storage\_id* id of the storage to add this new folder to. notice that you cannot mismatch storage id and parent id: they must both be on the same storage! Pass in 0 if you want to create this folder on the default storage.

**Returns:**

id to new folder or 0 if an error occurred

References **LIBMTP\_mtpdevice\_struct::params**, and **LIBMTP\_mtpdevice\_struct::usbinfo**.

**void LIBMTP\_destroy\_folder\_t** (**LIBMTP\_folder\_t \*** **folder**)

This recursively deletes the memory for a folder structure. This shall typically be called on a top-level folder list to destroy the entire folder tree.

**Parameters:**

*folder* folder structure to destroy

**See also:**

**LIBMTP\_new\_folder\_t()**

References **LIBMTP\_folder\_struct::child**, **LIBMTP\_destroy\_folder\_t()**, **LIBMTP\_folder\_struct::name**, and **LIBMTP\_folder\_struct::sibling**.

Referenced by **LIBMTP\_destroy\_folder\_t()**.

**LIBMTP\_folder\_t\* LIBMTP\_Find\_Folder** (**LIBMTP\_folder\_t \*** **folderlist**, **uint32\_t** **id**)

Helper function. Returns a folder structure for a specified id.

**Parameters:**

*folderlist* list of folders to search *id* of folder to look for

**Returns:**

a folder or NULL if not found

References **LIBMTP\_folder\_struct::child**, **LIBMTP\_folder\_struct::folder\_id**, **LIBMTP\_Find\_Folder()**, and **LIBMTP\_folder\_struct::sibling**.

Referenced by **LIBMTP\_Find\_Folder()**.



**LIBMTP\_folder\_t\* LIBMTP\_Get\_Folder\_List (LIBMTP\_mtpdevice\_t \* device)**

This returns a list of all folders available on the current MTP device.

**Parameters:**

*device* a pointer to the device to get the folder listing for.

**Returns:**

a list of folders

References LIBMTP\_Get\_Folder\_List\_For\_Storage().

**LIBMTP\_folder\_t\* LIBMTP\_Get\_Folder\_List\_For\_Storage (LIBMTP\_mtpdevice\_t \* device, uint32\_t const storage)**

This returns a list of all folders available on the current MTP device.

**Parameters:**

*device* a pointer to the device to get the folder listing for.

*storage* a storage ID to get the folder list from

**Returns:**

a list of folders

References LIBMTP\_mtpdevice\_struct::params.

Referenced by LIBMTP\_Get\_Folder\_List().

**LIBMTP\_folder\_t\* LIBMTP\_new\_folder\_t (void)**

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

**Returns:**

a pointer to the newly allocated folder structure.

**See also:**

**LIBMTP\_destroy\_folder\_t()**

References LIBMTP\_folder\_struct::folder\_id.

**int LIBMTP\_Set\_Folder\_Name (LIBMTP\_mtpdevice\_t \* device, LIBMTP\_folder\_t \* folder, const char \* newname)**

This function renames a single folder. This simply means that the PTP\_OPC\_ObjectFileName property is updated, if this is supported by the device.

**Parameters:**

*device* a pointer to the device that contains the file.

*folder* the folder metadata of the folder 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.

**Author**

Generated automatically by Doxygen for libmtp from the source code.

