

1 Introduction

TransferManager manages the download tasks by HTTP/BT/Xunlei.

TransferManger adds/creates a task to download resource by HTTP/BT/Xunlei. If a task is created OK, TranferManager returns task id for App to control the task. After the task is created, App can start, pause or close it. File to describe the status of the tasks is saved to continue downloading resource. Tasks are loaded from the file when system reboots.

NOTE: Streaming Manager is not encapsulated by TransferManager now.

2 Interface

The main interfaces of TranferManager Module are listed below.

transfer_mgr_init

Synopsis

`xfer_errid_t transfer_mgr_init();`

Description

Initialize the TransferManager Module.

Parameters

Return

error id

transfer_mgr_fini

Synopsis

`Void transfer_mgr_fini();`

Description

Initialize the TransferManager Module.

Parameters

void

Return

void

transfer_mgr_module_register

Synopsis

```
xfer_errid_t transfer_mgr_module_register(transfer_module_type_e type, void *param);
```

Description

Register the TransferManager Module.

Parameters

transfer_module_type_e type: [IN]transfer module type.

void *param: [IN] the param needed by transfer module register, such as license that is needed for Xunlei module to authenticate.

Return

error id

transfer_mgr_module_unregister

Synopsis

```
xfer_errid_t transfer_mgr_module_unregister(xfer_module_type_e type);
```

Description

Unregister the TransferManager Module.

Parameters

xfer_module_type_e type: [IN]transfer module type.

Return

error id

transfer_mgr_load

Synopsis

```
xfer_errid_t transfer_mgr_load(const char* path);
```

Description

Load tasks from the task information file.

NOTE: If there are not tmp files (http.aml, pend.aml and thunder.aml) in the path, they will be created.

Parameters

char *path: [IN] directory where the task information file is saved.

Return

error id

transfer_mgr_task_add

Synopsis

```
transfer_mgr_task_add( const char*      source_path,
                      const char*      saved_path,
                      transfer_type_e    type,
                      int               mode,
                      const char*       post_buff,
                      transfer_complete_cb complete_callback,
                      transfer_taskid_t* taskid);
```

Description

Add/Create a task to download the resource described by @source_path.

Parameters

char *source_path: [IN] the URL of the downloaded file or the name of torrent file.
char *save_path: [IN] the path to save the file downloaded.
transfer_type_e type: [IN] It may be BT/HTTP/FTP/XUNLEI, etc.
int mode:
const char* post_buff:
transfer_complete_cb complete_callback:
transfer_taskid_t* taskid: [OUT] If add/create task OK, ID of the task is returned by it.

Return

error id

transfer_mgr_task_add_bt

Synopsis

```
transfer_mgr_task_add_bt( const char*      source_url,
                          const char*      seed_path,
                          const char*      saved_path,
                          const char*      file_name,
                          UINT32*          file_index_array,
                          UINT32           file_num,
                          transfer_taskid_t* taskid);
```

Description

Add/Create a bt task.

Parameters

const char *source_path: [IN] the source URL of the downloaded file.
const char *seed_path: [IN] the name of bt torrent file.
const char *save_path: [IN] the path to save the file downloaded.

const char *file_name: [IN] the file name.
UINT32* [IN] the array of selected download file index in torrent.
UINT32: [IN] the count of selected download files.
transfer_taskid_t* taskid: [OUT] If add/create task OK, ID of the task is returned by it.

Return

error id

transfer_mgr_task_add_vod

Synopsis

```
xfer_errid_t transfer_mgr_task_add_vod( const char*      cid,  
                                       const char*      gcid,  
                                       const char*      saved_path,  
                                       u64_t           task_len,  
                                       transfer_type_e   type,  
                                       const char*      file_name,  
                                       transfer_taskid_t* taskid);
```

Description

Add/Create a vod task.

Parameters

const char *cid: [IN] the cid of th Kankan task.
const char *gcid: [IN] the gcid of the Kankan task.
const char *save_path: [IN] the path to save the file downloaded.
u64_t task_len: [IN] the length of the Kankan task.
transfer_type_e type : [IN] the type of transfer. should be XFER_TASK_XLGCID.
const char *file_name: [IN] the file name.
transfer_taskid_t* taskid: [OUT] If add/create task OK, ID of the task is returned by it.

Return

error id

transfer_mgr_task_append_http_header

Synopsis

```
transfer_mgr_task_append_http_header (transfer_taskid_t taskid, char *header);
```

Description

Append a http header.

This function just can be called before the task starts, and the task type must be XFER_TASK_HTTP.

Parameters

transfer_taskid_t taskid: [IN] id of the task
char *header: [IN] the http headers that will be appended. The parameter header could be like this: "Accept-Language: zh-cn".

Return

error id

transfer_mgr_task_start

Synopsis

```
xfer_errid_t transfer_mgr_task_start(transfer_taskid_t taskid);
```

Description

Start the download task specified by @taskid.

Parameters

transfer_taskid_t taskid: [IN] id of the task to start

Return

error id

transfer_mgr_task_pause

Synopsis

```
xfer_errid_t transfer_mgr_task_pause(transfer_taskid_t taskid);
```

Description

Pause the download task specified by @taskid.

Parameters

transfer_taskid_t taskid: [IN] id of the task to pause

Return

error id

transfer_mgr_task_close

Synopsis

```
xfer_errid_t transfer_mgr_task_close(transfer_taskid_t taskid);
```

Description

Close the download task specified by @taskid.

Parameters

transfer_taskid_t taskid: [IN] id of the task to close.

Return

error id

transfer_mgr_task_close_all

Synopsis

```
xfer_errid_t transfer_mgr_task_close_all(transfer_module_type_e type);
```

Description

Close all download tasks specified by transfer_module_type_e type .

Parameters

transfer_module_type_e type: [IN] type of tasks.

Return

error id

transfer_mgr_getopt

Synopsis

```
xfer_errid_t transfer_mgr_getopt(transfer_options_e opt, transfer_type_e type, int * result);
```

Description

Get option for giving type defined by @type.

Option	HTTP	BT	XUNLEI
DOWNLOAD_SPEED_LIMIT	Support	Support	Support
UPLOAD_SPEED_LIMIT	-----	Support	Support
MAX_TASKS	-----	-----	Support
MAX_CONNECTIONS_PER_TASK	-----	-----	Support
SEED_FORMAT	-----	-----	Support
LICENSE	-----	-----	Support
DLOPT_HTTP_MAX_REDIRS	Support	-----	-----
DLOPT_HTTP_MAX_TIMEOUT	Support	-----	-----
DLOPT_HTTP_MAX_CONNTIMEOUT	Support	-----	-----

Parameters

transfer_options_e opt: [IN] Option to get.

transfer_type_e type: [IN] Type of Download, May be BT/HTTP/XUNLEI

int * result: [OUT] value of option

Return

error id

transfer_mgr_setopt

Synopsis

```
xfer_errid_t transfer_mgr_setopt(transfer_options_e opt, transfer_type_e type, int value);
```

Description

Set option for giving type defined by @type.

Parameters

transfer_options_e opt: [IN] Option to set.
transfer_type_e type: [IN] Type of Download, May be BT/HTTP/XUNLEI
int value: [IN] value of option

Return

error id

transfer_mgr_get_task_list

Synopsis

```
xfer_errid_t transfer_mgr_get_task_list(transfer_task_list_t ** tasklist);
```

Description

get the task list. Notice the task list is a duplicated copy so that there is no thread conflict

Parameters

transfer_task_list_t ** tasklist: [OUT] structure to describe the task list. It is allocated by the API. It is the caller's responsibility to free the structure.

Return

error id

transfer_mgr_release_task_list

Synopsis

```
void transfer_mgr_release_task_list(transfer_task_list_t * tasklist);
```

Description

Release the object pointed to by@tasklist.

Parameters

transfer_task_list_t ** tasklist: [IN] structure to describe the task list.

Return

Void

transfer_mgr_get_task_info

Synopsis

```
xfer_errid_t transfer_mgr_get_task_info(transfer_taskid_t taskid, transfer_task_info_t** taskinfo);
```

Description

Get task information, including name of the task, path to save file downloaded, task URL, task status, download speed, size of the resource to download, etc. See the definition of the structure *task_info_t*.

Parameters

transfer_taskid_t taskid: [IN] task ID
transfer_task_info_t** taskinfo: [OUT] structure to describe the task information. It is allocated by the API. It is the caller's responsibility to free the structure.

Return

error id

transfer_mgr_release_task_info

Synopsis

```
void transfer_mgr_release_task_info(transfer_task_info_t* taskinfo);
```

Description

Release the object pointed to by @taskinfo.

Parameters

transfer_task_info_t* taskinfo: [IN] pointer to the structure to release.

Return

error id

transfer_mgr_get_seed_info

Synopsis

```
xfer_errid_t transfer_mgr_get_seed_info(const char *seed_path, torrent_seed_info_t** seed_info);
```

Description

Get seed information. See the definition of the structure *torrent_seed_info_t*.

Parameters

const char *seed_path: [IN] torrent seed path
torrent_seed_info_t** seed_info: [OUT] structure to describe the seed information. It is allocated by the API. It is the caller's responsibility to free the structure.

Return

error id

transfer_mgr_release_seed_info

Synopsis

```
xfer_errid_t transfer_mgr_get_seed_info(const char *seed_path, torrent_seed_info_t** seed_info);
```

Description

Get seed information. See the definition of the structure **torrent_seed_info_t**.

Parameters

torrent_seed_info_t* seed_info: [IN] pointer to the structure to release.

Return

error id

transfer_mgr_get_task_src_url

Synopsis

```
xfer_errid_t transfer_mgr_get_task_src_url(transfer_taskid_t taskid, char* src_url);
```

Description

Get task source url.

Parameters

transfer_taskid_t taskid: [IN]task id.

char* src_url: [OUT]source url.

Return

error id

transfer_mgr_set_task_src_url

Synopsis

```
xfer_errid_t transfer_mgr_set_task_src_url(transfer_taskid_t taskid, char* src_url);
```

Description

Set task source url.

Parameters

transfer_taskid_t taskid: [IN]task id.

char* src_url: [IN]source url.

Return

error id

transfer_mgr_get_kankan_buffer_percent

Synopsis

```
transfer_mgr_get_kankan_buffer_percent (transfer_taskid_t taskid, int *percent);
```

Description

Get the buffer percent of Kankan task.

Parameters

transfer_taskid_t taskid: [IN]task id.
int* percent: [OUT] buffer percent.

Return

error id

transfer_mgr_set_path_prefix

Synopsis

```
transfer_mgr_set_path_prefix (const char* path);
```

Description

Set relative path prefix.

Parameters

Const char *path: [IN] file path. Must add "/" at the end of this path.

Return

error id