BitTorrent Software Development Kit

Version 2.0

Contents

1 Overview

1.1 Introduction

The BitTorrent Software Development Kit (SDK) is designed for use in embedded systems, including: network-area storage devices (NASs) and set-top boxes. It provides a state-of-the-art implementation of the BitTorrent protocol and a full-featured web-based user interface in a small footprint. In addition, the BitTorrent SDK aims to be highly-portable to a large number of devices and be easy to use both for integrators and end-users.

1.2 Features

The BitTorrent SDK is a full implementation of the official BitTorrent protocol. Features include:

- Distributed hash table (DHT)
- UPnP port mapping
- NAT-PMP port mapping

- Upload rate limiting
- Download rate limiting
- Configurable limit on number of simultaneously uploading peers
- Incremental file allocation
- Block level piece picking
- Separate threads for file-check and download
- Single thread and single port for multiple torrent downloads
- BitTorrent extension protocol
- Multitracker extension support
- Fair trade extension
- Compact tracker extension
- Fast resume
- Queuing of torrent file-check if fast resume not possible
- HTTP seed support
- Resumption of partial downloads from other BitTorrent clients
- File-sizes greater than 2GB
- Selective download of multi-file torrents
- IPv6
- High performance network stack

Additionally, the BitTorrent SDK inludes a full-featured web-based user interface that is fully customizable and extensible by licencees.

2 Getting Started

The BitTorrent SDK consist of:

- an executable daemon (`bt`) that implements BitTorrent services and is used through an HTTP-based application programming interface (API);
- a `btdog` watchdog process that monitors `bt` process and restarts it in case it crashes. This improves reliability of the system.
- a reference implementation of a web-based user interface (UI) for managing BitTorrent downloads through a web browser.

In order to integrate the SDK into your product, you must:

- make sure that the `bt` process is always running;
- install the web-based user interface, which consists of HTML, CSS and JavaScript files to be served either by a stand-alone web server (like Apache) or directly from the `bt` process.

The web-based UI is provided as a reference implementation. You may use it as-is, customize its look-and-feel by modifying CSS styles, or extend it to provide new functionality or tighter integration with your device or other applications.

Alternatively, you may build a completely new UI using the BitTorrent SDK API.

2.1 SDK Usage

There are 2 ways to deploy SDK on the device:

- run `bt` process under `btdog` supervision as: `btdog bt -daemon-loop` (where `bt` is the name of the process to launch and `-daemon-loop` is the argument to `bt` process). `btdog` is a watchdog process that will restart `bt` process in case it exits for any reson.
- run `bt -daemon` which starts `bt` process in Unix daemon mode.

For the reason of reliability, we strongly recommend using `btdog`.

At startup the `bt` executable looks for `btsettings.txt` which allows the behaviour of the SDK to be customized by changing the values of certain settings. The format of this file is as follows:

- each setting is on a separate line;
- each line consists of colon-separated name of the setting and its value;
- any line whose first non-whitespace character is `#` is a comment.

For example, a file that sets two values and includes one comment might look like:

```
# This is a comment
bind_port: 9388
max_total_connections: 600
```

For a complete list of application settings, see Application Settings.

3 API Documentation

3.1 Overview

The API provides a simple remote procedure call (RPC) interface to the BitTorrent SDK. All API calls are HTTP GET or POST requests. Some API calls return nothing, others return a result. When a result is returned, it may be returned in any one of three formats selectable at call time:

- JSON
- XML
- bencoded

JSON format is recommended for JavaScript applications and is the default return type; XML is a widely-used, standards-based markup language, and finally, bencoded results are small and easy to parse and, therefore, suitable for low-level languages like C, C++, C#, and Java.

JSON is the default return type. Other formats can be requested by appending the `format` argument to the query, with value equal to `xml` (for XML format) or `benc` (for bencoded format).

For example, the query `/api/app-settings-get` returns the current settings in JSON format, while the query `/api/app-settings-get?format=xml` returns them in XML format.

All results are returned as a dictionaries of key-value pairs. Future versions of the BitTorrent SDK may extend API results with new key-value pairs. Consequently, applications built upon the SDK should ignore unknown keys.

Successful API calls return 200 (OK) HTTP status response.

3.2 API calls

3.2.1 Getting Application Settings

`/api/app-settings-get[?format=$format]`

Returns application settings. For a complete list of settings see Application Settings.

Sample JSON result:

```
{"settings" :
 {"auto_bandwidth_management" : 1,
  "bind_port" : 6881,
  "conns_per_torrent" : 30,
  "max_dl_rate" : -1,
  "max_total_connections" : 400,
  "max_ul_rate" : -1,
  "max_ul_rate_seed" : -1,
  "seed_ratio" : 0,
  "seed_time" : 0,
  "ul_slots_per_torrent" : 4}}
```

Sample XML result:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
<settings>
 <auto_bandwidth_management>1</auto_bandwidth_management>
 <bind_port>6881</bind_port>
 <conns_per_torrent>30</conns_per_torrent>
 <max_dl_rate>-1</max_dl_rate>
 <max_total_connections>400</max_total_connections>
 <max_ul_rate>-1</max_ul_rate>
 <max_ul_rate_seed>-1</max_ul_rate_seed>
 <seed_ratio>0</seed_ratio>
 <seed_time>0</seed_time>
 <ul_slots_per_torrent>4</ul_slots_per_torrent>
</settings>
</result>
```

3.2.2 Setting Application Settings

`/api/app-settings-set?$name1=$val1[&$name2=$val2...]`

Sets one or more application settings. Silently ignores unknown setting names and invalid setting values.

Returns: nothing.

3.2.3 Getting Information About Managed Torrents

`/api/torrents-get[?format=$format][&hash=$infohash1][&hash=$infohash]`

> Returns a list of torrents and information about them. If no `hash` argument is provided, returns information for all torrents. If one or more `hash` arguments are provided, returns information only for torrents with those infohashes. Invalid `hash` arguments are silently ignored.
>
> Returns an empty result if there are no torrents or none of the `hash` arguments is valid.

The information that is returned for each torrent is as follows:

hash (*string*): Infohash of the torrent.

caption (*string*): Name of the torrent.

size (*integer*): Size (in bytes) of the torrent (combined size of all files in the torrent).

done (*integer*): Number of bytes downloaded so far. If `done` is equal to `size`, the torrent has been downloaded completely.

dl_rate (*integer*): Current download rate in bytes per second.

ul_rate (*integer*): Current upload rate in bytes per second.

payload_download (*integer*): Total number of bytes downloaded in a current session (since starting the application).

payload_upload (*integer*): Total number of bytes uploaded in a current session (since starting the application).

peers_total (*integer*): Total number of peers for this torrent.

peers_connected (*integer*): Number of peers to which we are connected.

seeds_connected (*integer*): Total number of seeds (peers that have the complete torrent) for this torrent.

seeds_total (*integer*): Number of seeds to which we are connected.

private (*Boolean*): 1 (true) if this is a private torrent, 0 (false) otherwise.

state (*string*): State can be one of the following: “queued_for_checking”, “checking_files”, “connecting_to_tracker”, “downloading”, “finished”, “seeding”, or “allocating”.

stopped (*Boolean*): 1 (true) if the torrent is stopped, 0 (false) otherwise.

distributed_copies (*string*): String representation of a floating point number representing the total number of distributed copies. Values lower than 1.0 mean that a torrent cannot be fully download from current peers.

max_dl_rate (*integer*): Maximum download rate for this torrent in kilobytes per second. -1 means unlimited. Default value: -1.

max_ul_rate (*integer*): Maximum upload rate for this torrent in kilobytes per second. -1 means unlimited. Default value: -1.

max_uploads (*integer*): Maximum number of uploads (peers that can download at the same time) for this torrent. -1 means unlimited. Default value: 4.

max_connections (*integer*): Default value: 60.

An example JSON output:


```
{
  "torrents" : [ {
      "caption" : "Fedora-8-Live-KDE-i686",
      "distributed_copies" : "-1",
      "dl_rate" : 0,
      "done" : 732189042,
      "hash" : "5de112084598ee6b93f3b0602477d7efd1b47632",
      "max_connections" : 60,
      "max_dl_rate" : -1,
      "max_ul_rate" : -1,
      "max_uploads" : 4,
      "payload_download" : 0,
      "payload_upload" : 0,
      "peers_connected" : 0,
      "peers_total" : 230,
      "private" : 0,
      "seeds_connected" : 0,
      "seeds_total" : 198,
      "size" : 732189042,
      "state" : "seeding",
      "stopped" : 1,
      "ul_rate" : 0
  }, {
      "caption" : "mspevack-ohio-linux-fest-2007.ogg",
      "distributed_copies" : "-1",
      "dl_rate" : 0,
      "done" : 351013357,
      "hash" : "d6e183d38da44d03ba56bb3018fb5c75d4de9917",
```

```
        "max_connections" : 60,
        "max_dl_rate" : -1,
        "max_ul_rate" : -1,
        "max_uploads" : 4,
        "payload_download" : 0,
        "payload_upload" : 0,
        "peers_connected" : 0,
        "peers_total" : 11,
        "private" : 0,
        "seeds_connected" : 0,
        "seeds_total" : 10,
        "size" : 351013357,
        "state" : "seeding",
        "stopped" : 0,
        "ul_rate" : 0
    } ]
}
```

An example XML output:

```
<result>
<torrents>
 <item>
   <caption>Fedora-8-Live-KDE-i686</caption>
   <distributed_copies>-1</distributed_copies>
   <dl_rate>0</dl_rate>
   <done>732189042</done>
   <hash>5de112084598ee6b93f3b0602477d7efd1b47632</hash>
   <max_connections>60</max_connections>
   <max_dl_rate>-1</max_dl_rate>
   <max_ul_rate>-1</max_ul_rate>
   <max_uploads>4</max_uploads>
   <payload_download>0</payload_download>
   <payload_upload>114688</payload_upload>
   <peers_connected>2</peers_connected>
   <peers_total>228</peers_total>
   <private>0</private>
   <seeds_connected>0</seeds_connected>
   <seeds_total>195</seeds_total>
   <size>732189042</size>
   <state>seeding</state>
   <stopped>0</stopped>
   <ul_rate>1638</ul_rate></item>
 <item>
```

```
    <caption>mspevack-ohio-linux-fest-2007.ogg</caption>
    <distributed_copies>-1</distributed_copies>
    <dl_rate>0</dl_rate>
    <done>351013357</done>
    <hash>d6e183d38da44d03ba56bb3018fb5c75d4de9917</hash>
    <max_connections>60</max_connections>
    <max_dl_rate>-1</max_dl_rate>
    <max_ul_rate>-1</max_ul_rate>
    <max_uploads>4</max_uploads>
    <payload_download>0</payload_download>
    <payload_upload>0</payload_upload>
    <peers_connected>0</peers_connected>
    <peers_total>9</peers_total>
    <private>0</private>
    <seeds_connected>0</seeds_connected>
    <seeds_total>8</seeds_total>
    <size>351013357</size>
    <state>seeding</state>
    <stopped>0</stopped>
    <ul_rate>0</ul_rate></item>
  </torrents>
 </result>
```

3.2.4 Setting the Properties of a Torrent

`/api/torrent-set-props?hash=$infohash&$name1=$val1[&$name2=$val2...]`

Set one or more properties of a given torrent. Invalid properties and invalid values are silently ignored.

Returns: nothing.

The torrent properties that may be set are:

max_dl_rate (*integer*): Maximum download rate for this torrent in kilobytes per second. -1 means unlimited. Default value: -1.

max_ul_rate (*integer*): Maximum upload rate for this torrent in kilobytes per second. -1 means unlimited. Default value: -1.

max_uploads (*integer*): Maximum number of uploads (peers that can download at the same time) for this torrent. -1 means unlimited. Default value: 4.

max_connections (*integer*): Default value: 60.

3.2.5 Starting a Torrent

`/api/torrent-start?hash=$infohash`

Starts the torrent with the given infohash. Infohashes can be obtained with the `/api/torrents-get` call. Does nothing if the torrent is already started.

Returns: nothing.

3.2.6 Stopping a Torrent

`/api/torrent-stop?hash=$infohash`

Stops the torrent with the given infohash. Infohashes can be obtained with the `/api/torrents-get` call. Does nothing if the torrent is already stopped.

Returns: nothing.

3.2.7 Adding a Torrent via URL

`/api/torrent-add?url=$url[&start=yes]`

Adds the torrent with the given URL, where `url` is a link to a torrent file accessible via HTTP. Invalid or inaccessible URLs are silently ignored.

If the optional `start` argument is `yes`, the torrent will start downloading automatically. Otherwise it must started explicitly with the `/api/torrent-start` API call.

Returns: nothing.

This call is asynchronous. It will return immediately without waiting to finish downloading the torrent file. This means that there might be a lag between when the `/api/torrent-add` call finishes and when the torrent appears in the list returned by `api/torrents-get`.

Example:

`/api/torrent-add?url=http://torrent.fedoraproject.org/torrents/Fedora-8-Live-i686.torrent`

3.2.8 Adding a Torrent File

`/api/torrent-add[&start=yes]`

Add the torrent file provided by HTTP POST data. The POSTed torrent file data are expected in multipart/form-data encoded format. This call is synchronous.

Returns: a short HTML document invoking JavaScript functions to support web-based user interfaces (see below).

If the torrent file was valid, returns:

```
<html>
  <body>
  <script type=\"text/javascript\">
  if (window.parent.apiTorrentAddFinishedOk) win-
dow.parent.apiTorrentAddFinishedOk();
  </script>
  </body>
</html>
```

If the torrent file was invalid, returns:

```
<html>
  <body>
  <script type=\"text/javascript\">
  if (window.parent.apiTorrentAddFailed) win-
dow.parent.apiTorrentAddFailed();
  </script>
  </body>
</html>
```

These responses may be ignored but help in building web-based interfaces. By implementing `apiTorrentAddFailed()` and `apiTorrentAddFinishedOk()` JavaScript functions, a web-based UI can notify the user if the call failed or succeeded.

In a web-based UI, it is necessary to use a hidden IFRAME in the invoking page to capture the response. This is shown in the following example:

```
<form id="uploadFile" method="POST" action="/api/torrent-
add" target="uploadFrame" enctype="multipart/form-data">
```

```
        <input id="uploadFile" type="file" size="50" name="fileEl">
        <input type="submit" value="Submit"/>
</form>
<iframe id="uploadFrame" name="uploadFrame" style="width:0px; height:0px; bor-
der: 0px"></iframe>
```

3.2.9 Removing a Torrent

`/api/torrent-remove?hash=$infohash[&delete-torrent=yes][&delete-data=yes]`

Removes the torrent with the given infohash. If the optional `delete-torrent` argument is `yes`, also deletes the torrent file. If the optional `delete-data` argument is `yes`, also deletes the data for this torrent.

Returns: nothing.

It is recommended that `delete-torrent` and `delete-data` always be set to `yes` if the torrent has not yet been fully downloaded.

3.2.10 Getting a List of Files in a Torrent

`/api/torrent-get-files?[?format=$format][&hash=$infohash1][&hash=$infohash]`

Returns a list of files in a torrent and their properties. If no `hash` argument is provided, returns information for all torrents. If one or more `hash` arguments are provided, returns information only for torrents with those infohashes. Invalid `hash` arguments are silently ignored.

Returns an empty result if there are no torrents or none of the `hash` arguments is valid.

The properties returned for each file are:

id (*integer*): A unique id for the file (needed for some other API calls).

name (*string*): Name of the file (might include a directory).

size (*integer*): Size of the file.

done (*integer*): How much of the file has been downloaded.

pri (*integer*): Download priority of the file. Files with higher priority might be downloaded faster than other files.

0 means low, 1 means default (medium), 2 means high. -1 means: do not download at all.

Example JSON response:

```
{
  "torrents" : {
      "1606c977d334534b1bf12149399529a4c89b33d0" : [ {
          "done" : 719859712,
          "id" : 0,
          "name" : "Fedora-7-KDE-Live-i686\\Fedora-7-KDE-Live-
i686.iso",
          "pri" : 1,
          "size" : 719859712
      }, {
          "done" : 370,
          "id" : 1,
          "name" : "Fedora-7-KDE-Live-i686\\SHA1SUM",
          "pri" : 1,
          "size" : 370
      } ],
      "37a721f52791ddd0eac551e380e4716690cee6f9" : [ {
          "done" : 732942336,
          "id" : 0,
          "name" : "Fedora-8-Live-ppc\\Fedora-8-Live-ppc.iso",
          "pri" : 1,
          "size" : 732942336
      }, {
          "done" : 300,
          "id" : 1,
          "name" : "Fedora-8-Live-ppc\\SHA1SUM",
          "pri" : 1,
          "size" : 300
      } ]
  }
}
```

Example XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
<torrents>
 <1606c977d334534b1bf12149399529a4c89b33d0>
   <item>
     <done>719859712</done>
     <id>0</id>
```

```
      <name>Fedora-7-KDE-Live-i686%5CFedora-7-KDE-Live-i686.iso</name>
      <pri>0</pri>
      <size>719859712</size></item>
    <item>
      <done>370</done>
      <id>1</id>
      <name>Fedora-7-KDE-Live-i686%5CSHA1SUM</name>
      <pri>0</pri>
      <size>370</size></item>
  </1606c977d334534b1bf12149399529a4c89b33d0>
  <37a721f52791ddd0eac551e380e4716690cee6f9>
    <item>
      <done>732942336</done>
      <id>0</id>
      <name>Fedora-8-Live-ppc%5CFedora-8-Live-ppc.iso</name>
      <pri>0</pri>
      <size>732942336</size></item>
    <item>
      <done>300</done>
      <id>1</id>
      <name>Fedora-8-Live-ppc%5CSHA1SUM</name>
      <pri>0</pri>
      <size>300</size></item>
   </37a721f52791ddd0eac551e380e4716690cee6f9>
 </torrents>
 </result>
```

3.2.11 Setting the Download Priority of a File

`/api/torrent-file-set-priority?hash=$infohash&id=$id&pri=$pri`

Set the download priority of the file specified by the given infohash and file id. The priority argument may be:

-1: do not download the file
0: low priority
1: default (medium) priority
2: high priority

Silently ignores invalid `hash` or `id` or `pri`.

Returns: nothing.

3.2.12 Get File Content

`/api/torrent-file-get?hash=$infohash&id=$id`

Returns the content of the file specified by the given infohash and file `id`. File `id` may be obtained from the `/api/torrent-get-files` call.

If a `hash` or `id` argument is invalid, returns 404 HTTP response.

4 Web UI

The web-based UI is provided as a reference implementation. You may use it as-is, customize its look-and-feel by modifying CSS styles, or extend it to provide new functionality or tighter integration with your device.

Alternatively, you may build a completely new UI on top of the BitTorrent SDK API.

The BitTorrent SDK provides a Downloads view, from which torrent downloads may be managed, and a Settings page, where several settings may be customized.

The BitTorrent SDK actually comes with two alternative Download views:

`fe01.html` a power-user view similar to PC-based clients such as uTorrent.

`fe02.html` a simpler user interface, hiding most detail, appropriate for new or non-technical users

4.1 Installing the Reference UI

The reference UI consists of HTML, CSS, JavaScript and image files. The files can be served by the `bt` process or a standard HTTP server (e.g. Apache).

BitTorrent does not provide end-user customer support for the SDK. The support link that appears on the web UI is initially set to `http://sdk_licencee_support_page/` and must be changed by licencees to a customer support site or knowledge-base operated by the licencee.

4.1.1 Serving Files Directly from the `bt` Process

The `bt` process can serve as an embedded HTTP server and serve all the HTML, CSS and JavaScript files. The advantage of this scenario is that no other software is required; you need

only configured the `bt` process properly. The disadvantage is that, if you already have an HTTP server running, the `bt` process must use a different HTTP port.

4.1.2 Serving Files from an HTTP Server

If you already have an HTTP server running, the HTTP server can serve all HTML, CSS and JavaScript files, while the `bt` process provides data and functionality via its HTTP API.

One problem with this approach is that the requests to the `bt` process are done via JavaScript `XMLHTTPRequest` calls which are restricted by the same-origin policy of most browsers. That means that an `XMLHTTPRequest` call can only issue requests to exactly the same server from which HTML pages are served. Since the HTTP server (serving HTTP files) runs on a different port than the `bt` process responding to the `XMLHTTPRequest` calls, this violates same-origin policy.

There are several possible workarounds. One workaround, specific to the Apache HTTP server, is to configure the server to proxy `XMLHTTPRequest` calls in a way that is invisible to the HTTP server.

First, you must make sure that Apache is compiled with `mod_rewrite`, `mod_proxy` and `mod_proxy_http`. These modules also must be loaded, which is done by adding the following to `httpd.conf`:

```
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

Assuming that both the HTTP server and the `bt` process are running on localhost (127.0.0.1) and the `bt` process is serving HTTP requests on port 8080, the following lines in `.htaccess` file will correctly configure proxying:

```
RewriteEngine On
RewriteBase /
RewriteRule ^api/(.*)$ http://127.0.0.1:8080/api/$1 [P]
```

4.1.3 Using the First Reference UI (`fe01.html`)

First reference web UI consists of following files:

- fe01.html
- btbase.css

- btlteIE6.css
- btsettings.html
- btfe00.css
- btsdk.js
- btuicommon.js
- img

You need to rename `fe01.html` to `index.html` and change href references in `btsettings.html` from `./fe01.html` to `index.html`

4.1.4 Using the Second Reference UI (`fe02.html`)

Second reference web UI consists of following files:

- fe02.html
- btbase.css
- btlteIE6.css
- btsettings.html
- btfe00.css
- btsdk.js
- btuicommon.js
- img

You need to rename `fe02.html` to `index.html` and change href references in `btsettings.html` from `./fe01.html` to `index.html`

5 Application Settings

Settings fall into two categories:

- internal settings, whose values can only be set through `btsettings.txt` file;
- regular settings, whose values can be set through `btsettings.txt` file or the `/api/app-settings-set` RPC API call.

A setting can be of one of three types:

- *string*
- *integer*
- *Boolean* value (`1` for true and `0` for false)

5.1 Internal Settings

bind_ip (*string*): IP address to use for socket connections. If not provided, a default IP address will be used. We do not recommend changing this value.

webui_port (*integer*): Default value: 8080. Port number where the `bt` process accepts HTTP RPC API calls. If the `bt` process also serves HTML files (see `webui_server_files` setting), also the port of HTTP server.

webui_serve_files (*boolean*): Default value: true. If true, the `bt` process will act as an HTTP server and serve HTML, CSS and JavaScript files needed for webui. When set to false, the web UI files will be served by a stand-alone HTTP server (like Apache).

webui_dir_files (*string*): Default value: "webui". Name of the directory with HTML, CSS and JavaScript files that constitute web UI. It can be an absolute path (recommended) or set relative to current directory of `bt` process.

webui_root (*string*): Default value: "/". If the `bt` process also acts as an HTTP server, prefix for web UI. E.g. if `webui_root` is "/foo/", a request for "/foo/bar.html" will make `bt` process return `bar.html` from `webui_dir_files` directory.

dir_active (*string*): Default value: "./". Directory in which currently downloaded data is saved. Can be an absolute path or a relative path. If it is a relative path, the value is relative to `dir_root` or the current working directory if `dir_root` is not defined or an empty string. It is recommended that this directory be hidden from users (i.e. not exported through Samba).

dir_completed (*string*): Default value: "". Directory where a completed downloads are stored. If empty string, value of `dir_active` is used. This must be a path that is accesible to users (e.g. exported through Samba). It also has to be on the same volume as `dir_active`.

dir_torrent_files (*string*): Default value: "". Directory where torrent files are stored. If empty string, value of `dir_active` is used. It is recommended that this directory be hidden from users (i.e. not exported through Samba).

upnp (*boolean*): Default value: true. If true, UPNP functionality for mapping ports is used by `bt`. We recommend setting its value to true.

natpmp (*boolean*): If true, NAT-PMP functionality for mapping ports is used by `bt`. Default value: true. We recommend settings its value to true.

lsd (*boolean*): Default value: true. If true, Local Service Discovery is enabled. We recommend settings its value to true.

dht (*boolean*): Default value: true. If true, Distributed Hash Table extension is enabled. We recommend settings its value to true.

pex (*boolean*): Default value: true. If true, Peer Exchange extension is enabled. We recommend settings its value to true.

dir_root (*string*): Default value: "". If not empty, **dir_active**, **dir_completed**, and **dir_torrent_files** are relative to this directory.

5.2 Regular Settings

bind_port (*integer*): Default value: 6881. Port used for BitTorrent protocol. This can be any value in the range 1025-65000.

max_ul_rate (*integer*): Default value: -1. Maximum total upload rate in bytes per second. -1 means unlimited. We recommend setting it to -1.

max_ul_rate_seed (*integer*): Default value: -1. Maximum per-torrent upload rate when seeding, in bytes per second. -1 means unlimited. We recomment setting it to -1.

ul_slots_per_torrent (*integer*): Default value: 4. Maximum number of peers that can download a given torrent at the same time.

conns_per_torrent (*integer*): Default value: 50. Maximium number of connections for a given torrent.

max_total_connections (*integer*): Default value: 200. Maximum number of connection opened at the same time.

auto_bandwidth_management (*boolean*): Default value: true. If true, upload bandwidth is automatically throttled in order to not impact other applications using TCP/IP.

max_dl_rate (*integer*): Default value: -1. Maximum total download rate in bytes per second. -1 means unlimited. We recommend setting it to -1.

seed_ratio (*integer*): Default value: 0. Seed ratio in percent (%) (e.g. 100 means 100%). If not 0, seeding will stop after reaching this upload/download ratio.

seed_time (*integer*): Default value: 0. Time after which seeding will stop, in seconds. 0 means it wont stop.

6 Legal Notices

BitTorrent is a trademark or registered trademark of BitTorrent, Inc. used only under license.

This SDK contains the libtorrent library, which is subject to the following:

> Original libtorrent library, Copyright (c) 2007, Arvid Norberg. All rights reserved. Modifications to libtorrent library Copyright (c) 2007, BitTorrent, Inc. All rights reserved.
>
> Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
>
> - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
> - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
> - Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
>
> THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.