Transmission Documentation

Release 2.92+

Transmission Project

December 13, 2016

1 Daemon **1**
1.1 Daemon Installation Guide - Unix . . . 1
1.2 Daemon Configuration Guide - Unix . . . 1
1.3 Daemon Remote Management - Unix . . . 2
1.4 Scripts . . . 2
1.5 Default Locations . . . 4
1.6 Supported OS Versions . . . 4

2 Web Interface **7**
2.1 Enabling the Web Interface . . . 7
2.2 Accessing the Web Interface . . . 7

3 Building From Source **9**
3.1 Getting the Source . . . 9
3.2 On Mac OS X . . . 9
3.3 On Unix . . . 10
3.4 On Windows . . . 12
3.5 Switches . . . 13

4 Blocklists **15**
4.1 What blocklist does Transmission Use? . . . 15
4.2 Adding Other Blocklists . . . 15
4.3 Using Blocklists in transmission-daemon . . . 15

5 Configuration Files **17**
5.1 Locations . . . 17
5.2 Files . . . 18
5.3 Legacy Versions of Transmission . . . 18

6 Editing Configuration Files **19**
6.1 GTK+ / Qt / Daemon / CLI . . . 19
6.2 Mac OS X . . . 26

7 Environment Variables **27**
7.1 Transmission-Specific Variables . . . 27
7.2 Standard Variables Used By Transmission . . . 27
7.3 Standard Variables Used By Other Tools . . . 28

8 RPC Protocol Specification **29**

9 Indices and tables 31

CHAPTER 1

Daemon

1.1 Daemon Installation Guide - Unix

1.1.1 Generic Instructions

Installation is platform and distribution specific. The general steps are:

1. Download the Binary or Source packages for **`transmission-daemon`**
2. Install the Binary Package or [wiki:Building#OnUnix Build from Source]
3. Create a user for Transmission to run in (recommended: "transmission")
4. Create a start/stop script for your platform (e.g. Scripts)
5. Identify where the key configuration files are located (i.e. `settings.json`). Although the Transmission project has default locations, some distributions may relocate these files to comply with distribution specific guidelines. Distributions usually provide tools that can show where package files are installed.
6. Edit the configuration file for your site requirements (download directories etc.)

`daemon-install` covers the above setup steps in more detail.

See also:

Transmission Configuration (based on Nexenta but of general use)

1.1.2 OS-Specific Instructions

Specific daemon installation instructions have been provided for the following systems:

- `daemon-on-gentoo`
- `daemon-on-nslu2`
- `daemon-on-solaris`

If you can give a clear description of the steps required for an undocumented platform, and provide a sample start/stop script, please [/newticket open a ticket] (set Component to Wiki).

1.2 Daemon Configuration Guide - Unix

See also:

- [HeadlessUsage/General General Daemon Installation] covers the above setup steps in more detail.
- Transmission Configuration (based on Nexenta but of general use)
- Scripts
- `transmission-daemon(1) manual page`
- `transmission-remote(1) manual page`
- `transmission-cli(1) manual page`

Configuration information:

- Configuration Files
- Editing Configuration Files
- Environment Variables

1.3 Daemon Remote Management - Unix

1.4 Scripts

1.4.1 Introduction

Thanks to the powerful RPC, **`transmission-remote`** can talk to any client that has the RPC enabled. This means that a script written using **`transmission-remote`** or RPC can, without rewrite, comunicate with all the Transmission clients: GTK+ client, Mac client and the daemon.

Mac OS users may wonder wether there will be Applescript scripts, the answer is *no*. Although Applescript is a nice technology, it's a pain to implement. However, Mac OS X is a Unix after all, so any script you find here will also work on the Mac. Even from within Applescript, you can run these scripts by typing:

```
do shell script "path/to/script"
```

1.4.2 How-To

If you are interested at writing scripts for Transmission, have a look at the following pages:

- [wiki:man Transmission man pages]
- Configuration Files
- Editing Configuration Files
- Environment Variables
- RPC Protocol Specification

For those who need more information how to use the scripts, have a look at the following links:

- Cron How-To: Run scripts at a regular interval

1.4.3 Start/Stop Scripts

- [wiki:Scripts/initd init.d script] (Debian, Ubuntu and BSD deratives)
- [wiki:Scripts/runscript runscript] (Gentoo and other runscript-compatible systems)

1.4.4 Scripts On Torrent Completion

Transmission can be set to invoke a script when downloads complete. The environment variables supported are:

TR_APP_VERSION

> **Todo**
>
> Add description.

TR_TIME_LOCALTIME

> **Todo**
>
> Add description.

TR_TORRENT_DIR

> **Todo**
>
> Add description.

TR_TORRENT_HASH

> **Todo**
>
> Add description.

TR_TORRENT_ID

> **Todo**
>
> Add description.

TR_TORRENT_NAME

> **Todo**
>
> Add description.

[https://trac.transmissionbt.com/browser/trunk/extras/send-email-when-torrent-done.sh Here is an example script] that sends an email when a torrent finishes.

1.4.5 Obsolete Scripts

Functionality of these scripts has been implemented in libtransmission and is thus available in all clients.

- [wiki:Scripts/EmailNotifier Email Notification Script]
- [wiki:Scripts/BlockListUpdater Block List Updater]
- [wiki:Scripts/Watchdog Watch Directory Script]
- [wiki:Scripts/Scheduler Bandwidth Scheduler]

1.4.6 Contributed Scripts

Tomas Carnecky (a.k.a. wereHamster) is maintaining a set of scripts in his github repository.

Falk Husemann (a.k.a. hxgn) is maintaining scripts in his blog.

oguz wrote on his blog a PHP script to stop Transmission after it finishes downloading and seeding.

Scripts which have not yet been ported and may not work with the latest version:

- http://pastie.org/338556: Python - Fetch new torrents from tvrss
- http://pastie.org/338555: PHP - Stop finished torrents
- http://pastie.org/443058: Perl - Network traffic graph, based on rrdtool (example: http://skitch.com/werehamster/bmjg8/bittorrent-traffic)
- http://transmission.pastebin.com/QzVxQDtM: Bash - (cron)script to keep a maximum number of torrents running; starting and pausing torrents as necessary
- https://github.com/jaboto/Transmission-script - (cron)script set network limits according to the number of clients in the network

1.5 Default Locations

These tend to be OS/distribution specific.

Users of **Debian**-based distributions (e.g. **Ubuntu**) should read `daemon-on-debian`.

Users of **Arch Linux**-based distributions should read `daemon-on-arch`.

1.6 Supported OS Versions

Transmission Daemon should be able to run on any Unix based platform.

The minimium recommended system specifications are:

- 200 MHz CPU
- 64 MB RAM

Transmission Daemon has been tested on:

- Mac OS X 10.7 and up
- Linux (Arch, Debian, Fedora, Gentoo, Mandriva, OpenSUSE, Pardus, Puppy, Slackware, Ubuntu)
- BSD (FreeBSD, NetBSD, OpenBSD)

- Solaris 10
- Windows Vista and up

Transmission Daemon is also run on embedded devices including:

- Networked Media Tank
- ReadyNAS
- Synology
- WD MyBook
- D-Link DNS-323 & CH3SNAS
- IP Box 9000

CHAPTER 2

Web Interface

A Web Interface is built into all Transmission flavours, enabling them to be controlled remotely.

2.1 Enabling the Web Interface

Refer to the **Remote Management** links in the *User Documentation* for your client to find out how to enable the Web Interface.

2.2 Accessing the Web Interface

Once enabled, open a web browser and direct it to http://<IP address of machine running Transmission>:9091/. If web browser and Transmission Daemon are on the same machine you can use http://127.0.0.1:9091/.

9091 is the default remote control port specified in *Transmission configuration*.

CHAPTER 3

Building From Source

If you are searching for a HOWTO, covering a specific distribution or device (NAS, router, ...), have a look at the [HeadlessUsage "Running Transmission on a headless machine"] page

3.1 Getting the Source

3.1.1 Recommended

Source code for official releases can be found on our download page.

3.1.2 Experimental

Automated source code tarballs including the newest code are now available, too!

If you want to check out the source code yourself from svn, open a terminal window and type:

```
$ svn co svn://svn.transmissionbt.com/Transmission/trunk Transmission
```

3.2 On Mac OS X

Transmission has an Xcode project file (Transmission.xcodeproj) for building in Xcode. Make sure you have this software:

- OS X 10.8 or newer
- OS X 10.8 SDK
- Xcode 4.4 or newer

Building the project on Mac requires the source to be retrieved from SVN. Pre-packaged source code will not compile.

If building from source is too daunting for you, check out the nightly builds.

3.3 On Unix

3.3.1 Prerequisites

Ubuntu

On Ubuntu, you can install the required development tools with this command:

```
$ sudo apt-get install build-essential automake autoconf libtool pkg-config intltool libcurl4-openssl
```

''After you install those you can skip [#Buildingfromatarball to this section].''

Debian Squeeze

Sometimes you have a need to stay current with upstream releases, even though you would like to rely on the stability of your base distribution. Here is how this can be accomplished in "quick and dirty" fashion. Lines started with a # are to be executed as root, lines starting with $ can be run as a regular user.

1. Dependencies

First let us install every dependency Transmission needs and for which there is a usable version in the Debian repository.

```
# apt-get install ca-certificates libcurl4-openssl-dev libssl-dev pkg-config build-essential checkins
```

2. libevent

Traditionally, libevent is also needed, but Transmission depends on version numbers only rarely found in Debian. So let us start by compiling libevent in a directory of your choice. Browse to http://libevent.org/ and get the latest version.

```
$ cd /var/tmp
$ wget https://github.com/downloads/libevent/libevent/libevent-2.0.18-stable.tar.gz
$ tar xzf libevent-2.0.18-stable.tar.gz
$ cd libevent-2.0.18-stable
$ CFLAGS="-Os -march=native" ./configure && make
```

Now, we would really like to be able to upgrade to a new version in the future, so there should be a mechanism other than the classic **`make install`** which keeps count of what went where (and ideally this is not a piece of paper). So we build a very simple Debian package from the compiled files and install it. Basically you just enter the following command and hit return until a nice text message tells you that all is done.

```
# checkinstall
```

3. Transmission

Now we need to prepare Transmission for compilation by configuring the source, the same as with libevent.

```
$ cd /var/tmp
$ wget http://download-origin.transmissionbt.com/files/transmission-2.51.tar.bz2
$ tar xjf transmission-2.51.tar.bz2
$ cd transmission-2.51
# CFLAGS="-Os -march=native" ./configure && make && checkinstall
```

Thanks to josen at http://falkhusemann.de/blog/2012/05/compiling-transmission-bittorrent-for-debiand/ for the original Debian Squeeze howto section.

CentOS 5.4

The packages you need are:

- gcc
- gcc-c++
- m4
- make
- automake
- libtool
- gettext
- openssl-devel

Or simply run the following command:

```
$ yum install gcc gcc-c++ m4 make automake libtool gettext openssl-devel
```

However, Transmission needs other packages unavailable in {{{yum}}}:

- pkg-config
- libcurl
- intltool

Before building Transmission, you need to set the pkgconfig environment setting:

```
$ export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
```

''After you install those you can skip [#Buildingfromatarball to this section].''

Normal

If this is your first time compiling on Unix, you'll need a few basic tools:

- gcc
- libtool
- gettext 0.14.1 or newer
- intltool 0.40 or newer

If you're planning to build from SVN:

- automake 1.9 or newer
- autoconf 2.54 or newer

Once you've got the basics out of the way, here are the libraries that Transmission needs to have in order to build:

- OpenSSL 0.9.8 or newer, preferably ssl or gnutls support.
- libcurl 7.16.3 or newer
- GTK+ 2.6 or newer (only needed by the GTK+ gui)
- libnotify 0.4.4 (optional, and only needed by the GTK+ gui)
- DBUS 0.70 (optional, and only needed by the GTK+ gui)

RPM users

You'll also need to install the corresponding ''-devel'' packages.

3.3.2 Building from a tarball

```
$ tar xvjf transmission-1.76.tar.bz2
$ cd transmission-1.76
$ ./configure -q && make -s
$ su # if necessary for the next line
$ make install
```

3.3.3 Building from an SVN snapshot

First Time

```
$ svn co svn://svn.transmissionbt.com/Transmission/trunk Transmission
$ cd Transmission
$ ./autogen.sh && make -s
$ su # if necessary for the next line
$ make install
```

Updating

```
$ cd Transmission
$ make clean
$ svn up
$ make -s
$ su # if necessary for the next line
$ make install
```

3.4 On Windows

For Windows XP and above there are several choices:

3.4.1 Cygwin environment

With Cygwin installed, the CLI tools (transmission-remote, transmissioncli, etc.) and the daemon can be built easily.

No patches needed(*), all the recent versions of Transmission built almost out-of-the-box (you need to install the prerequisites), and the CLI tools work better under Cygwin that those built with MinGW.

(*) At the release time of version 2.0, **libevent** is not bundled and it's also not in Cygwin distribution (but was added later)... so you need to build it (which is as easy as ./configure, make install). To build transmission you may need to add LDFLAGS="-L/usr/local/lib" to the configure script (LIBEVENT_LIBS doesn't seem to work when it comes to build all the test programs). Additionally **libutp** needs deleting -ansi on the Makefile.

With version 2.51 miniupnpc fails to build, see http://miniupnp.tuxfamily.org/forum/viewtopic.php?t=1130.

Version 2.80 breaks building on Cygwin, adding this https://github.com/adaptivecomputing/torque/blob/master/src/resmom/cygwin/quota file to Cygwin's `/usr/include/sys` solves the problem. This is no longer needed after version 2.82 (Cygwin added the header).

Version 2.81 with the above workaround needs a one line patch, see ticket #5692.

Version 2.82, same as 2.81.

Version 2.83, no need to add quota.h, Cygwin added it.

3.4.2 Native Windows

With a MinGW development environment, the Gtk and the Qt GUI applications can be built. The CLI tools can also be built, and in general work fine, but may fail if you use foreign characters as parameters (MinGW uses latin1 in parameters).

The procedure: wiki:BuildingTransmissionQtWindows

3.5 Switches

The transmission {{{./configure}}} (or {{{./autogen.sh}}}) script allows you to switch on/off certain parts. To use these, you'll either use `--enable-*` or `--disable-*`. E.g. to disable the GTK client: `--disable-gtk`.

The switches that are available are:

- **gtk** = enables GTK+ client (default)
- **daemon** = enables transmission-daemon and *-remote client (default)
- **cli** = enables cli client (default. deprecated, consider using the daemon)
- **libnotify** = enables lib notify (default)
- **nls** = enables native language support (default)
- **mac** = enables Mac client (default, if possible)
- **wx** = enables wxWidgets client (unsupported)
- **beos** = enables beos client (unsupported)

Note: `--disable-nls` removes the dependancy on gettext and intltool. It's designed for, and should only be used on, [HeadlessUsage embedded devices]. If you do have GTK+ installed on your box, you must also specify `--disable-gtk`.

CHAPTER 4

Blocklists

A Blocklist is third-party list of peer addresses to block. This can be used to block peers whose addresses are believed to belong to spyware or malware manufacturers.

4.1 What blocklist does Transmission Use?

Transmission supports the P2P Plaintext Format, which is used by PeerGuardian, Bluetack, Vuze, ProtoWall, and KTorrent, and the DAT format, which was originally made popular by eMule.

The Transmission Project does not evaluate or endorse any specific blocklists. If you do not know what blocklist to use, you might read about some third-party blocklists and evaluate them on your own.

If *Enable automatic updates* is enabled, Transmission will periodically refresh its copy of your blocklist from your specified URL.

When you press the *Update Blocklist* button, Transmission will download a new copy of your blocklist.

4.2 Adding Other Blocklists

Transmission stores blocklists in a folder named `blocklists` in its configuration folder (see *Configuration Files*).

In that directory, files ending in ".bin" are blocklists that Transmission has parsed into a binary format suitable for quick lookups. When Transmission starts, it scans this directory for files not ending in ".bin" and tries to parse them. So to add another blocklist, all you have to do is put it in this directory and restart Transmission. Text and gzip formats are supported.

4.3 Using Blocklists in transmission-daemon

`transmission-daemon` doesn't have an *Update Blocklist* button, so its users have two options. They can either copy blocklists from **`transmission-gtk`**'s directory to **`transmission-daemon`**'s directory, or they can download a blocklist by hand, uncompress it, and place it in the daemon's `blocklists` folder. In both cases, the daemon's `settings.json` file (see *Configuration Files*) will need to be edited to set *`blocklist-enabled`* to `true`.

Also in both cases, the daemon is unaware of blocklist updates. Only when it starts it creates new .bin files.

There is a third option: add the blocklist URL in `settings.json` (only one blocklist is allowed), and use **`transmission-remote`** to tell the daemon to update it periodically.

`settings.json` snippet:

```
{
    "blocklist-enabled": true,
    "blocklist-url": "http://www.example.com/blocklist",
}
```

update by hand example:

```
$ transmission-remote -n admin:password --blocklist-update
localhost:9091/transmission/rpc/ responded: "success"
```

CHAPTER 5

Configuration Files

5.1 Locations

5.1.1 Mac OS X Defaults

What	Where
Per-torrent settings	`$HOME/Library/Application Support/Transmission`
Application settings	`$HOME/Library/Preferences/org.m0k.transmission.plist`
Default download folder	`$HOME/Downloads`

5.1.2 *NIX Defaults

What	Where
GTK+/Qt client settings	`$HOME/.config/transmission`
Daemon settings	`$HOME/.config/transmission-daemon`
CLI settings	`$HOME/.config/transmission-cli`
Default download folder	`$HOME/Downloads`

Some Linux distributions' start scripts for transmission-daemon use different location. This varies by distribution (see e.g. `daemon-on-debian`), but two paths sometimes used are `/var/lib/transmission-daemon` and `/var/run/transmission`.

If you want to swap between the two applications, all you have to do is pass in a different config directory with the `-g` command-line option. For example, to have the daemon pick up where the GTK+ client left off, run:

```
$ transmission-daemon -g ~/.config/transmission
```

5.1.3 Windows

What	Where
Qt client settings	`%LOCALAPPDATA%/transmission`
Daemon settings	`%LOCALAPPDATA%/transmission-daemon`
Default download folder	`%LOCALAPPDATA%/Downloads`

5.1.4 Overriding the Defaults

The per-user configuration folder's location can be overridden by setting `TRANSMISSION_HOME` and/or other *environment variables*.

5.2 Files

The configuration folder typically has the following files:

`settings.json` This is a json-encoded file that holds all the client's settings and preferences. It's currently only used by the daemon, GTK+ and CLI clients. json was chosen because it's mostly human-readable and -writable, but also allows data hierarchies to be stored. See *Editing Configuration Files* on how to modify them.

`stats.json` This is a json-encoded file that holds session statistics such as running upload and download byte counts.

`torrents/` This subfolder holds the .torrent files that have been added to Transmission. The files in this folder are named with a combination of the torrent's name (to make it human-readable) and a portion of the torrent's SHA1 hash (to avoid filename collisions from similarly-named torrents).

`resume/` This subfolder holds .resume files that hold information about a particular torrent, such as which parts have been downloaded, the folder the downloaded data was stored in, and so on. These follow an identical naming scheme to the files in the torrents subfolder.

`blocklists/` This subfolder holds bluetack-formatted blocklists. Files ending in ".bin" are generated by Transmission as it parses a bluetack file and stores it into a binary format for faster lookups. On startup, Transmission will try to parse any non-".bin" file and generate a new blocklist from it, so you can have multiple blocklists just by copying new bluetack files into this location. See *Blocklists* for more information.

5.3 Legacy Versions of Transmission

Older, pre-XDG versions of transmission-gtk and transmission-daemon stored their settings in `$HOME/.transmission`. Newer releases try to automatically migrate these files to `$HOME/.config/transmission`.

CHAPTER 6

Editing Configuration Files

It's not always possible to set all configurations from the GUI, especially on the Daemon or the Web Interface. This guide will try to give an overview of how and what you can change. For the location of these files, look at the *Configuration Files* page.

Note: The client *should* be closed before making changes, otherwise settings will be reverted to it's previous state.

Some of Transmission's behavior can also be customized via *environment variables*.

6.1 GTK+ / Qt / Daemon / CLI

6.1.1 Overview

GTK+, Qt, CLI, and Daemon (both on a Mac and Linux) use a JSON formatted file, mainly because of its human readability.

6.1.2 Reload Settings

You can make the daemon reload the settings file by sending it the `SIGHUP` signal. Or, simply run either of the following commands:

```
$ killall -HUP transmission-da
```

Or:

```
$ pkill -HUP transmission-da
```

6.1.3 Formatting

Here is a sample of the three basic types, respectively Boolean, Number and String:

```
{
    "rpc-enabled": true,
    "peer-port": 51413,
    "rpc-whitelist": "127.0.0.1,192.168.*.*"
}
```

6.1.4 Options

Bandwidth

alt-speed-enabled : Boolean = false
Enable alternate Speed limits (a.k.a. "Turtle Mode").

> **Note:** Clicking the "Turtle" in the GUI when the *scheduler* is enabled will only temporarily remove the scheduled limit until the next cycle.

alt-speed-up : Number *(KB/s)* = 50
Alternate upload speed limit.

alt-speed-down : Number *(KB/s)* = 50
Alternate download speed limit.

speed-limit-down : Number *(KB/s)* = 100
Standard download speed limit (used if enabled).

speed-limit-down-enabled : Boolean = false
Flag to enable the use of `speed-limit-down`.

speed-limit-up : Number *(KB/s)* = 100
Standard upload speed limit (used if enabled). May need to be set for best performance.

speed-limit-up-enabled : Boolean = false
Flag to enable the use of `speed-limit-up`.

upload-slots-per-torrent : Number = 14

Blocklists

See also:

Blocklists

blocklist-url : String = "http://www.example.com/blocklist"

blocklist-enabled : Boolean = false

Files and Locations

download-dir : String = <platform-specific>
See *Locations* for default values.

incomplete-dir : String = <platform-specific>
Directory to keep files in until torrent is complete. See *Locations* for default values.

incomplete-dir-enabled : Boolean = false
When enabled, new torrents will download the files to `incomplete-dir`. When complete, the files will be moved to `download-dir`.

preallocation : Number = 1

Value	Meaning
0	Off
1	Fast
2	Full (slower but reduces disk fragmentation)

rename-partial-files : Boolean = true
Postfix partially downloaded files with ".part".

start-added-torrents : Boolean = true
Start torrents as soon as they are added.

trash-original-torrent-files : Boolean = false
Delete torrents added from the watch directory.

umask : Number = 18
Sets Transmission's file mode creation mask. See *umask(2)* for more information. Users who want their saved torrents to be world-writable may want to set this value to `0`. Bear in mind that the json markup language only accepts numbers in base 10, so the standard *umask(2)* octal notation `022` is written in settings.json as `18`.

watch-dir : String
Directory to watch for new .torrent files to autoload.

watch-dir-enabled : Boolean = false
Watch a directory for torrent files and add them to Transmission.

Note: When *watch-dir-enabled* is `true`, only the **transmission-daemon**, **transmission-gtk**, and **transmission-qt** applications will monitor *watch-dir* for new .torrent files and automatically load them.

Misc

cache-size-mb : Number *(MB)* = 4
Size to allocate for Transmission's memory cache. The cache is used to help batch disk IO together, so increasing the cache size can be used to reduce the number of disk reads and writes. Default is `2` if configured with `--enable-lightweight`.

dht-enabled : Boolean = true
Enable Distributed Hash Table (DHT).

encryption : Number = 1
Encryption preference. Encryption may help get around some ISP filtering, but at the cost of slightly higher CPU use.

Value	Meaning
0	Prefer unencrypted connections
1	Prefer encrypted connections
2	Require encrypted connections

lazy-bitfield-enabled : Boolean = true
May help get around some ISP filtering. Vuze specification.

lpd-enabled : Boolean = false
Enable Local Peer Discovery (LPD).

message-level : Number = 2
Set verbosity of transmission messages.

Value	Meaning
0	None
1	Error
2	Info
3	Debug

pex-enabled : Boolean = true
Enable Peer Exchange (PEX).

prefetch-enabled : Boolean = true
When enabled, Transmission will hint to the OS which piece data it's about to read from disk in order to satisfy requests from peers. On Linux, this is done by passing `POSIX_FADV_WILLNEED` to *`posix_fadvise(2)`*. On OS X, this is done by passing `F_RDADVISE` to *`fcntl(2)`*. This defaults to `false` if configured with `--enable-lightweight`.

scrape-paused-torrents-enabled : Boolean = true

script-torrent-done-enabled : Boolean = false
Run a script at torrent completion.

script-torrent-done-filename : String = ""
Path to script.

utp-enabled : Boolean = true
Enable Micro Transport Protocol (µTP).

Peers

bind-address-ipv4 : String = "0.0.0.0"
Where to listen for peer connections.

bind-address-ipv6 : String = "::"
Where to listen for peer connections.

peer-congestion-algorithm : String
This is documented on http://www.pps.jussieu.fr/~jch/software/bittorrent/tcp-congestion-control.html.

peer-id-ttl-hours : Number = 6
Recycle the peer id used for public torrents after *N* hours of use.

peer-limit-global : Number = 240
Maximum number of connected peers.

peer-limit-per-torrent : Number = 60
Maximum number of connected peers for an individual torrent.

peer-socket-tos : String = "default"
Set the Type-Of-Service (TOS) parameter for outgoing TCP packets. The value `"lowcost"` is recommended if you're using a smart router, and shouldn't harm in any case.

Value	Meaning
`"default"`	
`"lowcost"`	
`"throughput"`	
`"lowdelay"`	
`"reliability"`	

Peer Port

peer-port : Number = 51413
Port to listen on for incoming Peer connections.

peer-port-random-high : Number = 65535
Highest permitted value for a randomly assigned *`peer-port`*.

peer-port-random-low : Number = 1024
Lowest permitted value for a randomly assigned *peer-port*.

peer-port-random-on-start : Boolean = false
If true then assign a random *peer-port* between the *peer-port-random-low* and *peer-port-random-high* values.

port-forwarding-enabled : Boolean = true
Enable UPnP or NAT-PMP protocols to try and negotiate opening of selected *peer-port* with firewalls that support such negotiation. Security-conscious or advanced users may want to disable this feature.

Queuing

download-queue-enabled : Boolean = true
When `true`, Transmission will only download *download-queue-size* non-stalled torrents at once.

download-queue-size : Number = 5
See *download-queue-enabled*.

queue-stalled-enabled : Boolean = true
When `true`, torrents that have not shared data for *queue-stalled-minutes* are treated as "stalled" and are not counted against the *download-queue-size* and *seed-queue-size* limits.

queue-stalled-minutes : Number = 30
See *queue-stalled-enabled*.

seed-queue-enabled : Boolean = false
When `true`, Transmission will only seed *seed-queue-size* non-stalled torrents at once.

seed-queue-size : Number = 10
See *seed-queue-enabled*.

RPC

See also:

RPC Protocol Specification

rpc-authentication-required : Boolean = false
If set, then a username and password are required to access the remote control services.

rpc-bind-address : String = "0.0.0.0"
Where to listen for RPC connections.

rpc-enabled : Boolean = true
Enables remote control services.

rpc-password : String
The ssha1 encrypted password (starts with a `{`) needed for remote access. A new password can be entered via command line utilities or directly in plain text and will be replaced with the encrypted version when the configuration file is next saved.

rpc-port : Number = 9091
The port Transmission listens on for remote services.

rpc-url : String = "/transmission/"
Added in v2.2.

rpc-username : String
The username required to access remote services when *rpc-authentication-required* is enabled.

rpc-whitelist : String = "127.0.0.1"
Comma-delimited list of IP addresses from which remote control is permitted. Wildcards allowed using `"*"`. Example: `"127.0.0.*,192.168.*.*"`.

rpc-whitelist-enabled : Boolean = true
If enabled use *rpc-whitelist*. Other IP addresses will be denied remote access.

Scheduling

alt-speed-time-enabled : Boolean = false

> **Note:** When enabled, this will toggle the *alt-speed-enabled* setting.

alt-speed-time-begin : Number *(minutes from midnight)* = 540 *(9am)*

alt-speed-time-end : Number *(minutes from midnight)* = 1020 *(5pm)*

alt-speed-time-day : Number/bitfield = 127 *(all days)*
Start with `0`, then for each day you want the scheduler enabled, add:

Day	Decimal	Binary
Sunday	1	0000001
Monday	2	0000010
Tuesday	4	0000100
Wednesday	8	0001000
Thursday	16	0010000
Friday	32	0100000
Saturday	64	1000000

Examples:

Days	Decimal	Binary
Weekdays	62	0111110
Weekends	65	1000001
All days	127	1111111

idle-seeding-limit : Number = 30
Stop seeding after being idle for *N* minutes.

idle-seeding-limit-enabled : Boolean = false

ratio-limit : Number = 2.0
Ratio of uploads:downloads for a torrent before torrent is deemed complete. Ratio should be at least `1.0` for normal use, `2.0` is considered "good".

ratio-limit-enabled : Boolean = false
By default torrents will seed forever (i.e. Ratio → infinity).

6.1.5 Legacy Options

Only keys that differ from above are listed here. These options have been replaced in newer versions of Transmission.

2.31 (and older)

open-file-limit : Number = 32

1.5x (and older)

Bandwidth

download-limit : Number *(KB/s)* = 100

download-limit-enabled : Boolean = false

upload-limit : Number *(KB/s)* = 100

upload-limit-enabled : Boolean = false

Peer Port

peer-port-random-enabled : Boolean = false

1.4x (and older)

Proxy

proxy-authentication : String

proxy-authentication-required : Boolean = 0

proxy-port : Number = 80

proxy-server : String

proxy-server-enabled : Boolean = 0

proxy-type : Number = 0

Value	Meaning
0	HTTP
1	SOCKS4
2	SOCKS5

proxy-username : String

Peers

max-peers-global : Number = 240

max-peers-per-torrent : Number = 60

1.3x (and older)

RPC

See also:

RPC Protocol Specification

rpc-access-control-list : String = "+127.0.0.1"
Comma-delimited list of IP addresses prefixed with `"+"` or `"-"`. Wildcards allowed using `"*"`. Example: `"+127.0.0.*,-192.168.*.*"`.

6.2 Mac OS X

6.2.1 Overview

Mac OS X has a standardized way of saving user preferences files using XML format. These files are called plist (short for property list) files. Usually there is no need to modify these files directly, since Apple provided a *defaults(1)* command-line tool to reliably change settings. You do need to restart Transmission before these changes have effect.

In short:

- To set a key:

```
$ defaults write org.m0k.transmission <key> <value>
```

- To reset a key:

```
$ defaults delete org.m0k.transmission <key>
```

6.2.2 Options

PeerSocketTOS : Number = 0

CHAPTER 7

Environment Variables

Users can set environmental variables to override Transmission's default behavior and for debugging.

7.1 Transmission-Specific Variables

TRANSMISSION_HOME
If set, Transmission will look there for its settings instead of in the *default location*.

TRANSMISSION_WEB_HOME
If set, Transmission will look there for the Web Interface files, such as the javascript, html, and graphics files.

TR_CURL_VERBOSE
If set, debugging information for libcurl will be enabled. More information about libcurl's debugging mode is available here.

TR_DEBUG

Todo

Add description.

TR_DEBUG_FD
If set to an integer, that integer is treated as a file descriptor and very verbose debugging information is written to it. For example, here is how to turn on debugging and save it to a file named `runlog` when running Transmission from a bash shell:

```
$ export TR_DEBUG_FD=2
$ transmission 2>runlog
```

TR_DHT_VERBOSE
If set, then Transmission will log all of the DHT's activities in excrutiating detail to standard error.

7.2 Standard Variables Used By Transmission

- If *TRANSMISSION_WEB_HOME* is *not* set, non-Mac platforms will look for the Web Interface files in `XDG_DATA_HOME` and in `XDG_DATA_DIRS` as described in the XDG Base Directory Specification. `XDG_DATA_HOME` has a default value of `$HOME/.local/share/`.

- If `TRANSMISSION_HOME` is *not* set, Unix-based versions of Transmission will look for their settings in `$XDG_CONFIG_HOME/transmission/`. `XDG_CONFIG_HOME` has a default value of `$HOME/.config/`.
- If `HOME` is set, it's used in three ways:
  1. by the XDG (X Desktop Group) variables, as described above
  2. If `TRANSMISSION_HOME` is *not* set, Mac-based versions of Transmission will look for their settings in `$HOME/Library/Application Support/Transmission`
  3. `$HOME/Downloads` is the default download directory.

7.3 Standard Variables Used By Other Tools

Transmission uses the libcurl library for http- and https-based tracker announces and scrapes. Transmission doesn't support proxies, but libcurl itself honors a handful of environment variables to customize *its* proxy behavior.

CHAPTER 8

RPC Protocol Specification

CHAPTER 9

Indices and tables

- genindex
- modindex
- search

E

environment variable
- HOME, 28
- TR_APP_VERSION, 3
- TR_CURL_VERBOSE, 27
- TR_DEBUG, 27
- TR_DEBUG_FD, 27
- TR_DHT_VERBOSE, 27
- TR_TIME_LOCALTIME, 3
- TR_TORRENT_DIR, 3
- TR_TORRENT_HASH, 3
- TR_TORRENT_ID, 3
- TR_TORRENT_NAME, 3
- TRANSMISSION_HOME, 18, 27, 28
- TRANSMISSION_WEB_HOME, 27
- XDG_CONFIG_HOME, 28
- XDG_DATA_DIRS, 27
- XDG_DATA_HOME, 27

H

HOME, 28

O

org.m0k.transmission.plist
- PeerSocketTOS, 26

S

settings.json
- alt-speed-down, 20
- alt-speed-enabled, 20
- alt-speed-time-begin, 24
- alt-speed-time-day, 24
- alt-speed-time-enabled, 24
- alt-speed-time-end, 24
- alt-speed-up, 20
- bind-address-ipv4, 22
- bind-address-ipv6, 22
- blocklist-enabled, 20
- blocklist-url, 20
- cache-size-mb, 21
- dht-enabled, 21
- download-dir, 20
- download-limit, 25
- download-limit-enabled, 25
- download-queue-enabled, 23
- download-queue-size, 23
- encryption, 21
- idle-seeding-limit, 24
- idle-seeding-limit-enabled, 24
- incomplete-dir, 20
- incomplete-dir-enabled, 20
- lazy-bitfield-enabled, 21
- lpd-enabled, 21
- max-peers-global, 25
- max-peers-per-torrent, 25
- message-level, 21
- open-file-limit, 25
- peer-congestion-algorithm, 22
- peer-id-ttl-hours, 22
- peer-limit-global, 22
- peer-limit-per-torrent, 22
- peer-port, 22
- peer-port-random-enabled, 25
- peer-port-random-high, 22
- peer-port-random-low, 22
- peer-port-random-on-start, 23
- peer-socket-tos, 22
- pex-enabled, 22
- port-forwarding-enabled, 23
- preallocation, 20
- prefetch-enabled, 22
- proxy-authentication, 25
- proxy-authentication-required, 25
- proxy-port, 25
- proxy-server, 25
- proxy-server-enabled, 25
- proxy-type, 25
- proxy-username, 25
- queue-stalled-enabled, 23
- queue-stalled-minutes, 23
- ratio-limit, 24

ratio-limit-enabled, 24
rename-partial-files, 20
rpc-access-control-list, 25
rpc-authentication-required, 23
rpc-bind-address, 23
rpc-enabled, 23
rpc-password, 23
rpc-port, 23
rpc-url, 23
rpc-username, 23
rpc-whitelist, 23
rpc-whitelist-enabled, 24
scrape-paused-torrents-enabled, 22
script-torrent-done-enabled, 22
script-torrent-done-filename, 22
seed-queue-enabled, 23
seed-queue-size, 23
speed-limit-down, 20
speed-limit-down-enabled, 20
speed-limit-up, 20
speed-limit-up-enabled, 20
start-added-torrents, 21
trash-original-torrent-files, 21
umask, 21
upload-limit, 25
upload-limit-enabled, 25
upload-slots-per-torrent, 20
utp-enabled, 22
watch-dir, 21
watch-dir-enabled, 21

T

TRANSMISSION_HOME, 18, 28
TRANSMISSION_WEB_HOME, 27

X

XDG_CONFIG_HOME, 28
XDG_DATA_DIRS, 27
XDG_DATA_HOME, 27