

TClone IM client version 1.0 ReadMe Guide

1. REQUIREMENTS
2. UPLOAD
3. INSTALLATION
4. CONFIGURATION
5. RUNNING
6. DIRECTORY CONTENTS
7. SUPPORT

1. REQUIREMENTS

If the attached binary works by itself, only the libraries are required. Otherwise a new binary will have to be compiled, which will require development packages to be installed as well.

- mysql-client - a client library for MySQL (the database itself is not needed if it's on a different server)
- libpurple - a IM support library
- glib - a generic library with lots of useful functions (required by libpurple, so probably there will be no need to install it manually)
- libxml - for XML support (for example config file)
- libcurl - URL-manipulation library (for API support)

However to make things simpler, here are example ways of installing on some distributions. If the client needs to be compiled, then corresponding development packages will be needed too in most cases.

Red Hat

On all Red Hat based distributions (RHEL, Fedora, CentOS, etc.) - which is the most common case in commercial - the command to install packages is:

```
yum install _package_name_
```

for example:

```
yum install glib
```

Development packages usually have an additional "-devel" suffix. So for example installing a glib development package would be done using a command:

```
yum install glib-devel
```

Debian

On Debian based distributions (Debian, Gentoo, etc.) the command to install packages is "apt-get". Example command to install is:

```
apt-get install glib
```

Development packages in debian have the "-dev" suffix. example:

```
apt-get install glib-dev
```

Gentoo

Gentoo will probably not be the case on any popular web server, however for the sake of completion I'm including it here. Typical installation command is:

```
emerge glib
```

Since gentoo is a compiled distribution no development packages are needed.

Currently there is no separate package for libpurple, so to install it:

```
echo "net-im/pidgin -gtk" >> /etc/portage/packages.use  
emerge pidgin
```

About libpurple

On some distributions libpurple might not be readily available. To get it, go to

<http://pidgin.im/download>

And check installation instructions for correct procedure. Just make sure NOT to install "pidgin" just "purple" or "libpurple". Please also kindly make sure you have libpurple-devel installed as well

2. UPLOAD

Unzip the tclone.zip file and upload the entire folder to your server (any location) via SSH

3. INSTALLATION

Included in the file is a pre-compiled binary for 32bit systems. If this binary does not work (due to different ABI versions of used libraries), a new binary will have to be compiled.

The compilation process is rather standard for all unix-like systems, which means typing:

```
make
```

If everything goes correctly, a new tclone binary should be created.

Once the tclone binary is working, all source files are not needed, so they can be removed (this means all .c, .h and .o files).

This can be done manually, or by typing:

```
make rmsrc
```

INSTALLATION TROUBLESHOOTING

In some cases the installation might fail because proper libraries can't be found. This is most common with MySQL. To fix this, one should look for "libmysqlclient.so".

Once found, the Makefile should be edited, and in the line starting with "LDFLAGS=" a path to the file should be added, as such:

```
-L/path/to/mysql/
```

for instance if libmysqlclient.so is in

```
/usr/lib/mysql3/mysql/libmysqlclient.so
```

the line

```
LDFLAGS=`pkg-config --libs $(DEPS)` -lcurl -lmysqlclient
```

should be replaced with

```
LDFLAGS=`pkg-config --libs $(DEPS)` -L/usr/lib/mysql3/mysql/ -lcurl -lmysqlclient
```

4. CONFIGURATION

Inside the file tclone.xml is configuration for the client, containing the following options:

- disconnectTimeout - because of problems with some protocols (namely ICQ/AIM), accounts can't be reconnected immediately after they disconnect. That's why the connections are done in intervals. This value sets the number of seconds between connections. Higher numbers are safer, but can cause longer downtimes in case of disconnection.
- messageSendInterval - to avoid flooding, and reduce processor usage, new messages are sent in specific intervals too. This option sets the number of seconds between sending messages
- api - this sets the api options. Parameters are:
 - * url - the full address of the API (i.e <http://www.revou.com>)
 - * user - user to send messages as (keep it as 'admin')

- * key - API key for sending messages (keep it as '21232f297a57a5a743894a0e4a801fc3')
- database - Enter the mysql database information you have created for the ReVou software
- account - each item like this sets a new account to use. A list of example accounts is already provided to show what protocol/dbid values to use. If new protocols are added, then "protocol" is the name of libpurple plugin, while "dbid" is id used in the database.
*Please comment out any IM accounts that will not be in use.

5. RUNNING

Once configured, running the client means just executing the binary:

```
./tclient
```

This will run the program directly in the console, thus stopping on disconnection from the server. To make sure it doesn't get disconnected, it will have to be run in the background. To do it, two approaches are suggested:

***Note: Please run EITHER 1A or 1B**

#1A: Run it as a background shell process

This is usually done by redirecting the output, and detaching from the shell. This approach is probably simpler, but cuts out all output from the program, thus giving no chance to check on the status of the program.

If using bash or even sh (usually the default):

```
./tclient 2>&1 1>/dev/null &
```

or if using zsh, a better solution is (although the first one works too):

```
./tclient 2>&1 1>/dev/null &|
```

(note the '|' at the end)

If this approach is chosen, the client can be stopped by typing:

```
killall tclient
```

And can be checked if it's running by typing:

```
ps -A | grep tclient
```

#1B: use "screen" program

This approach requires additional software (namely the "screen" program), but gives more control over what's happening. Above all it allows checking of current status of the program - whether it's running and what messages it's giving.

First run screen:

```
screen
```

then run the client:

```
./tclient
```

and finally detach the screen:

```
[CTRL+A] [CTRL+D]
```

Once this is done, the screen program can be reattached, to check on the client by typing:

```
screen -D -R
```

Then it can be detached the same way as before:

```
[CTRL+A] [CTRL+D]
```

Or the "ps -A | grep tclient" can be used too without attaching to "screen".

*As for the screen - this is a standard program, practically present in every GNU/Linux distribution. For example in "CentOS" which is a flavour of Red Hat Linux - in case of THIS distribution, the `rpm -qa screen` is a command to CHECK if screen is installed. if it's NOT, then running "yum install screen" installs it

***Note: Whenever you edit the tclone.xml you need to restart the client using the following by running the following commands:**

1. Reattaching screen
screen -D -R
2. Stopping current client:
[CTRL+C]
3. Starting it again:
./tclient
4. And detaching screen:
[CTRL+A] [CTRL+D]

Screen in-depth explanation

The screen is a program that runs a separate console that does not get destroyed on disconnection from the server. Screen allows running a program with output redirected to console, and not risking it getting accidentally killed.

To command to start a new screen session is very simple:
screen

Or to attach to an already running session:
screen -R -D

Once in a screen session, it needs to be detached to go back to normal console. This can be achieved by pressing two key sequences:
[CTRL+A] and [CTRL+D]

If running tclone using the screen program, connecting to a screen session will show current output from the client. Since this mode is similar to the "pure" version (as it's still attached to a console), the client can be stopped by pressing [CTRL+C].

6. DIRECTORY CONTENTS

Following files are included:

- tclient - this is a dynamically compiled, 32bit binary of the client. Should work with the latest versions of needed libraries
- Makefile - instructions how to compile everything
- main.c - main libpurple functionality
- mysql.c, mysql.h - all functions that need database access (mostly reading user details and messages to send)
- api.c, api.h - API access routines
- options.c, options.h - config reader and parser
- functions.c, functions.h - other helpful functions
- config.h - some definitions
- tclone.xml - example configuration file (should be edited before running)

7. SUPPORT

Having problems with the server modules and TClone setup? ReVou offers hosting through JaguarPC.com that full supports our server requirements as well as TClone setup. ReVou customers will also get a 15% lifetime hosting discount with a purchase of our software. For more information on support, please visit www.revou.com/support.php