

# The Turtle F2F Client – User’s Manual

## 1 Introduction

This is the user’s manual for the Turtle client. The Turtle client can be used to connect to Turtle networks. A Turtle network is a *friend-to-friend* (a special case of *peer-to-peer*) overlay that can be used for safe and private data exchange. The idea for such a network is to facilitate the exchanging sensitive information that would be otherwise subject to censorship in an open environment, where parties providing/requesting such information would be exposed to various types of economical/legal/social pressure from the censors. The nature of the Turtle network ensures that such exchanges can take place without putting in danger any of the participating parties (information providers, consumers, or intermediaries).

The basic idea behind Turtle is to build a P2P overlay on top of pre-existent trust relationships among Turtle users. Each user acts as node in the overlay by running a copy of the Turtle client software. Turtle does not allow arbitrary nodes to connect and exchange information. Instead, each user establishes secure and authenticated channels with a limited number of other nodes controlled by people she trusts (friends). In the Turtle overlay, both queries and results move hop by hop; the net result is that at any moment, information is only exchanged between people that trust each other, and is always encrypted, so an adversary has no way to determine who is requesting/providing information, and what that information is about.

Section 2 describes the high-level structure of the Turtle client software. Section 3 of this manual describes how to install the Turtle client software on Linux platforms. Section 4 describes how to configure a Turtle node, and Section 5 explains how to connect to a Turtle network.

## 2 The Turtle client – components

The Turtle client distribution CD contains the following packages:

1. The giFT daemon – a generic P2P daemon supporting multiple P2P protocols. giFT is open-source third-party software (not developed by the Turtle team). For convenience, we provide *giFT.0.11.8.1* together with the Turtle distribution. You can also download giFT directly from the project homepage:  
<http://gift.sourceforge.net/>.
2. The Turtle network plugin for the giFT daemon (developed by the Turtle team).
3. A number of user interfaces for giFT:
  - *giFTcurs* – a cursor-based interface (no graphics). giFTcurs is open-source third-party software (not developed by the Turtle team). For convenience, we provide *giFTcurs.0.6.2* together with the Turtle distribution. You can also download *giFTcurs* directly from the project homepage:  
<http://www.nongnu.org/giftcurs/>.
  - *giFTtoxic* – a GTK2 based GUI, simple and small. giFTtoxic is open-source third-party software (not developed by the Turtle team). For convenience, we provide *giFTtoxic.0.0.10* together with the Turtle distribution. You can also download *giFTtoxic* directly from the project homepage:  
<http://www.sf.net/projects/gifttoxic>.

- Apollon – an elaborate GUI client for the KDE desktop environment. Apollon is open-source third-party software (not developed by the Turtle team). For convenience, we provide *Apollon.1.0.2.3* together with the Turtle distribution. You can also download *Apollon* directly from the project homepage: <http://apollon.sourceforge.net/>.

4. The Turtle configuration interface – a plugin for Apollon for configuring Turtle client nodes (developed by the Turtle team). *Important: The standard Apollon client does not directly support the Turtle configuration interface. In order to support Turtle-specific operations (adding friends, configuring keys, etc.) we have modified the Apollon client. In order to be able to use the Turtle configuration interface you need to install the Apollon distribution provided on this CD. If you choose to download Apollon from the net, you will have to configure Turtle via a configuration file.*
5. The OpenSSL cryptographic library – this is required for setting up secure channels between Turtle nodes. OpenSSL is open-source third-party software (not developed by the Turtle team). For convenience, we provide *OpenSSL.0.9.8* together with the Turtle distribution. You can also download OpenSSL directly from the project homepage: <http://www.openssl.org/>.

### 3 Installing the Turtle Client

#### 3.1 System requirements

In order to install the base Turtle system (which includes *.giFT* the *Turtle network plugin*, and simple *.giFT* clients, such as *giFTCurs* or *giFToxic*) you need a base Linux system and the development packages listed in Table 1

In addition to this, installing *Apollon* and the *Turtle configuration interface* requires the KDE desktop environment and development packages (version 3.4 or higher).

We have tested the Turtle distribution on the Linux distributions listed in Table 2.

Package	Minimum version
<i>autoconf</i>	2.59
<i>automake</i>	1.9.5
<i>gcc</i>	3.3.5
<i>glibc</i>	2.3.4
<i>libpng</i>	1.2.8
<i>libstdc++-devel</i>	3.3.5
<i>libtool</i>	1.5.14
<i>m4</i>	1.4.2
<i>make</i>	3.80
<i>zlib-devel</i>	1.2.2

**Table 1. Development packages required for installing the base Turtle system**

Linux distribution	Package sets selected
Fedora Core 5	Office Productivity Software Development
Suse 9.3	KDE C/C++ Compilers and Tools KDE Desktop KDE development
Mandiva 2006	Linux Development
Debian 3.1	Base Devel Perl Utils Admin KDE

**Table 2. Linux distributions on which the Turtle distribution was tested**

## 3.2 “All-in-one” install script

A “all-in-one” install script is provided on the Turtle CD, as well as on the Turtle Project Web page. At this moment, this script has been tested on Suse 9.3, Mandiva 2006, and Debian Linux distributions. If you are using any of these distributions, you should be able to install all required Turtle packages by copying the *turtleInstaller.sh* file to a temporary directory, and typing:

```
./turtleInstaller.sh
```

The script is a self extracting archive that contains all necessary packages, so you do not need to download anything else. Currently, our install script only supports the Apollon client. If you want to use any of the other clients, please follow the manual installation procedure (described next).

## 3.3 Manual installation procedure

These are the steps for installing Turtle on a target system:

1. Copy the *openssl-0.9.8.tar.gz*, *gift-0.11.8.1.tar.gz*, *gift-turtle-1.0.0.tar.gz*, *giFTcurs-0.6.2.tar.gz*, *giFToxic-0.0.10.tar.gz*, *apollon.kdevelop-1.0.2.3.tar.gz*, and *turtleui.kdevelop-0.2.tar.gz* archives from the distribution CD to a temporary directory.
2. If the *openssl* package is not installed on the target system, install it:

```
gunzip openssl-0.9.8.tar.gz
tar xvf openssl-0.9.8.tar
./config
make
su
make install
```

3. Install the *giFT* package:

```
gunzip gift-0.11.8.1.tar.gz
tar xvf gift-0.11.8.1.tar
./configure
make
su
make install
```

4. Install the Turtle plugin:

```
gunzip gift-turtle-1.0.0.tar.gz
tar xvf gift-turtle-1.0.0.tar
./configure
make
su
make install
```

5. Install a *giFT* client. You can choose between *giFTcurs*, *giFToxic*, or *Apollon*:

- for *giFTcurs*:

```
gunzip giFTcurs-0.6.2.tar.gz
tar xvf giFTcurs-0.6.2.tar
./configure
make
su
make install
```

- for *giFToxic*:

```
gunzip giFToxic-0.0.10.tar.gz
tar xvf giFToxic-0.0.10.tar
./configure
make
su
make install
```

- for *Apollon*:

```
gunzip apollon.kdevelop-1.0.2.3.tar.gz
tar xvf apollon.kdevelop-1.0.2.3.tar
./configure
make
su
make install
```

6. If you have chosen the *Apollon* client (the package that comes with the Turtle CD-ROM) you can also install the Turtle configuration interface:

```
gunzip turtleui.kdevelop-0.2.tar.gz
tar xvf turtleui.kdevelop-0.2.tar
./configure
make
su
make install
```

## 4 Configuring the Client

If you have installed the Turtle configuration interface (for the modified Apollon client), most of the configuration can be done from Apollon. Otherwise, you can configure your Turtle node by manually editing the configuration file.

### 4.1 Using the Turtle configuration interface

After the installation process is completed, you need to run `/usr/local/kde/bin/apollon`, which provides a first-time-run installation wizard. You will be required to provide a user name for your Turtle node (the name used by your friends to contact you). Once the main Apollon window is displayed, you can further configure your Turtle node by going to *Settings*→*Configure Apollon*→*Advanced*→*Turtle*→*Configure*. This will display the main Turtle configuration window. Fill in the fields there as follows:

1. Setup your Turtle node IP address. If you select *0.0.0.0* Turtle will be visible on all IP addresses on your host.
2. Setup the Turtle TCP port. By default this is 1391.

3. Setup the port where Turtle listens for key agreement requests from your friends. By default this is 4242.

4. Add and configure friend nodes:

- You can add new friend nodes by clicking on the *Add* button. This opens a new *Friend Node Settings* window, where you can fill in the details about the friend node:
  - The friend’s name – the name chosen by your friend for her Turtle node.
  - The friend’s node IP address.
  - The Turtle TCP port for the friend node.
  - The key agreement port for the friend node.
- Once a friend has been added, you can setup a shared key for authenticating to it, by selecting the friend from the “*Friend Nodes*” list in the main Turtle configuration window, and then clicking on the *Edit* button:
  - You can directly type in a shared key, agreed by out of band means (for example by meeting your friend in person). The shared key is represented as a 32 digits hex number. **Important:** Although not particularly user-friendly, for the time being, this is key agreement method should be used for the majority of Linux distributions (with the exception of Suse and Mandiva). The more user-friendly interactive key agreement mechanism (described next) is not yet stable, and is not guaranteed to work on all platforms.
  - You can generate a new key by performing an interactive key agreement protocol with your friend (**important:** only use this mechanism on Suse and Mandiva platforms. For all the other Linux platforms you need to type in the shared key). To start the interactive key agreement protocol, you click on the *Connect* button in the *Friend Node Settings* window. This will display a new *KeyAgreement Dialog* window. The interactive key agreement protocol takes place inside this window. The new key is created incrementally, through an interactive questions and answers session. You and your friend will in turn ask questions for which you both know the answer (Example - What is the name of our high school math teacher?). If the answers you provide match, they are computed into the shared key. The strength of the key depends on the number of questions/answers you submit. If you don’t know an answer to a question, push the *Don’t Know* button; your friend has the choice to send you another question, or abort the agreement. In order to have a secure key, it is important that answers depend on shared knowledge between you and your friend (it should be hard for a stranger to guess these answers!).
- Once a friend node has been added, and a shared key has been agreed upon, the Turtle software will automatically connect to the friend node, and establish a secure encrypted channel to it. This channel is then used for sending queries and receiving query results.

## 4.2 Manual configuration

If you have not installed the Turtle configuration interface, you need to configure Turtle manually. From the command line, type:

```
gift-turtle-ccfg.sh
```

This will execute the Turtle configuration script, which will generate a skeleton Turtle configuration file. This file is placed in `./giFT/Turtle/Turtle.conf`. You need to manually edit this file as follows:

1. Set your Turtle node name in the `[main]:myName` field.
2. Add your friend nodes in the `[neighbours]` section. Each friend is described by a set of *attribute-value* pairs of the form `friend_number/attribute_name = attribute_value`. For each friend you need to specify the following attributes:

- *name* – the name chosen by your friend for her Turtle node.
  - *address* – the IP address and port of your friend node (in the form *address:port*).
  - *kalp* – the key agreement port for the friend node.
  - *key* – the key shared between you and your friend (in this case you need to agree on this key by out of band means).
3. Set your node IP address and port in the *[tcp]:myAddress* field. The format is *address:port*. If you specify *0.0.0.0* as the IP address, Turtle will be visible on all IP addresses on your host.
  4. Set the port where Turtle listens for key agreement requests from your friends in the in the *[tcp]:keyAgreementListenerPort* field. *Note: in this case you will not be capable of running the interactive key agreement protocol (since you have not installed the Turtle configuration interface). This field is just a placeholder.*

### **4.3 Firewall configuration**

Turtle is a client-server application. Before running it, you should ensure that it is possible for other Turtle nodes (e.g. your friends) to connect to your node. If you run a firewall, you need to configure it in such a way that incoming connection requests for the Turtle Node Port and for the Turtle Key Agreement Listener Port (KALP) are allowed to pass through. This ensures that friend nodes can connect to your node. These two ports are the ones you set when you configured your Turtle node.

## **5 Connecting to the Turtle network**

Once you have added your friend nodes, the Turtle software will automatically connect to them. At this point, you can start searching the Turtle network. Select "Search" in the main Apollon window, type in the name of the file you want to find, and wait for the results!