# Urban Terror [Server] Guide

Sebastien Varrette

`Sebastien.Varrette@uni.lu`

Version 0.3.0-b8*

**Abstract**

This tutorial details the configuration of a Linux server for Urban Terror (UrT [4]), a free multiplayer first person shooter (FPS) based on the Quake 3 engine [3]. Urban Terror is very similar to CounterStrike and can be described as a Hollywood tactical shooter. Yet UrT is cross-platform meaning the software package exists for Windows, Linux and Mac OS X.

Whereas everybody can start a server from the client software, such approach does not authorize votes during the game, or statistics. Consequently, it is better to setup the server on a dedicated machine, as proposed in this document. In particular, apart from the UrT server setup, this tutorial details the installation of BigBrotherBot (B3) [2], a complete and total server administration package for online games (including UrT) and various plugins (for player statistics etc.).

The latest version of this tutorial (together with the LATEX sources of the document) are freely available on GitHub at the following address:

https://github.com/Falkor/urbanterror_guide.

---

*Draft version. Compiled time: 2012-07-26 11:40

# Contents

# 1 Introduction

Urban Terror (UrT for short) started as a realism mod for Quake III Arena, reusing most of the concepts of the famous game Counterstrike [1]. The approach is similar to the one proposed by the Tactical Ops[1] mod for Unreal Tournament. Yet with the release of the Quake III engine as an open-source software (through the `ioquake3` project [3]), Urban Terror is now completely free, stand-alone (*i.e.* it does not require to buy a copy of Quake III Arena) and, more importantly, cross-platform so it can run on Windows , Linux  and Mac OS X .

The software package in itself comes with two elements:

1. the client executable `ioUrbanTerror` for playing on a remote dedicated server. It is also possible to start a server at the same time you play to host a game and let other player join your game other the network;

2. a dedicated server executable `ioUrTded`.

Whereas the UrT client can start a server, you have access in this case to limited functionality (in particular no votes) and performances (as your machine also have to handle the graphics of your own game in addition to the server tasks), which justify the configuration of a dedicated server.

This guide is based on my own experience to share information about the configuration of a dedicated server for Urban Terror over a Linux machine (hosting a Debian distribution). I also detail the configuration of Big Brother Bot, a powerful and flexible server administration package for online games which support in particular UrT. It permits the use of various plugins, in particular `xlrstats`, which give access to statistics to the players of your server. The configuration of such plugins is also addressed by this document.

This document is not meant to be a reference guide (for this, just consider the official manuals that come with each package), I just found it would be interesting to centralize in a single document my notes for configuring each elements. It is released under the terms of the CC licence Creative Commons *Attribution-Noncommercial-Share Alike* 2.0 France, hoping other persons will help me to improve this document.

Your help, comments and feedback will be greatly appreciated ! Kindly address them by mail at `Sebastien.Varrette@uni.lu`. The LaTeX sources of this tutorial can also be obtained, as soon as you ask them by mail at the same address.

**Writing conventions.** When editing this document, I used several writing conventions summarized here:

- An item preceded by an OS symbol (, or ) qualify information specific to this OS;

---

[1]See `http://www.tactical-ops.de/`

- Command-lines are provided in a dedicated environment (grey boxes) prefixed by a prompt. This prompt can be of two types:

  1. [**user@host**]>

     This qualify a command that can be run with the rights of a normal user.

  2. [**root@host**]#

     This refers to a command to be run with superuser right (aka `root`) - use `su` or `sudo` commands to switch to this mod eventually (see the man pages for more information).

  Remember that this prompt is of course not part of the command provided.

**Organization.** This tutorial is organized as follows: §2 details the installation of the client software provided by the Urban Terror package. Information for all OS are given and a brief summary of some common game techniques are provided. Some advanced customization tricks (by adapting the `q3config.cfg` file) are also given. Then §3 expounds the configuration of a running dedicated server for Urban Terror on a Linux machine. In particular, some useful scripts (for instance to start/stop the server) are proposed. §4 is dedicated to Big Brother Bot (B3). At this level, the configuration of several plugins (`xlrstats` for instance) is detailed.

**Online repository.** The latest version of this document is available on my personal web site `http://varrette.gforge.uni.lu/q3ut4/urt_setup.pdf` The scripts provided in the appendix are available for download at the following url: `http://varrette.gforge.uni.lu/q3ut4/ConfigFiles/`

## 2 Client software

### 2.1 Installation

To play on UrT, just download the appropriate file (depending on your OS) from `http://www.urbanterror.net/`. When writing this tutorial, the last release of Urban Terror corresponded to v.4.1. Once done, just proceed to the installation:

You just have to launch the `UrbanTerror_41_FULL.exe` executable. Once this is done, you can launch Urban Terror through the shortcuts in your start menu and on your desktop if you chose to have them.

Under the other OS, you just have to download a zip file `UrbanTerror_41_FULL.zip`.

Get the unzipped folder `UrbanTerror` from the `Download` folder to `/Applications/`. You will then have to launch the executable `ioUrbanTerror.app`;

🐧 First unzip the file in the appropriate folder, typically `/usr/local/games`, then decide about using a executable you want to use (either 32 bits `ioUrbanTerror.i386` or 64 bits `ioUrbanTerror.x86_64`), make it executable and create the link to it. To make all those operations to use a 32 bits executable, you will typically run the following commands:

```
[user@host]> unzip UrbanTerror_41_FULL.zip
[root@host]# mv UrbanTerror /usr/local/games/urbanterror
[root@host]# cd /usr/local/games
[root@host]# chmod +x urbanterror/ioUrbanTerror.i386
[root@host]# ln −s urbanterror/ioUrbanTerror.i386 urt
```

You can now launch the game using the command:

```
[user@host]> /usr/local/games/urt
```

It is advised to create a dedicated group (`urt` typically) for the directory `/usr/local/games/urbanterror` and make the users of the computer supposed to launch the game (in this tutorial, the user with login `user`) member of this group:

```
[root@host]# addgroup urt
[root@host]# adduser user urt
[root@host]# chown −R :urt /usr/local/games/urbanterror
[root@host]# chmod −R g+w /usr/local/games/urbanterror
```

You are now more than encouraged to read the manual available on the following url: `http://www.urbanterror.net/new_urt_manual/`.

## 2.2   In-game technics

I really don't pretend to be an expert of the game yet I can provide some general hints and techniques that will make you fully enjoy this game..

**Choose your weapons and equipments carefully.** Your choices should be dictate by your playing style and the maps. You can choose up to three items, depending on how many weapons you have chosen to equip yourself with, and whether or not you have grenades. You will *always* carry a sidearm (*i.e.* a pistol, to be chosen between a Desert Eagle and a Beretta) and knives. The combinations of weapons, items and grenades are:

- 2 weapons, a sidearm, grenades and 1 item
- 2 weapons, a sidearm, 2 items
- 1 weapon, a sidearm, 3 items
- 1 weapon, a sidearm, grenades, 2 items

Don't hesitate to move on dead bodies to get grenades, ammos and eventually a more appropriate weapon than the one you hold.

**Know the maps.** Knowing them and the specific paths inside them will come with the time spent in the game and the ghosting of other players. It will make your movements more accurate, either to place in a strategic position or to move faster from one bomb point to another.

**Move cleverly and analyse your environment.** Don't be static and use the sound to locate your enemies. In addition, always keep in mind the position of your partners using the minimap to identifying enemy activities. Sprint to move faster throughout the map (see dedicated item), walk to stay silent when approaching a dangerous area and crouch to hide and/or be more precise when shooting.

**Shoot with automatic weapons cleverly.** If possible, aim before shooting (typically at the head). In all cases, no need to empty your magazine in a single shoot: the dispersion will probably make you bullets fail to reach your opponent. Instead, you should prefer successive short shots. Always remember to reload to be prepared to the next battle[2]. If your weapon magazine becomes empty during a confrontation, remember to switch to your sidearm. Finally, knives are quite deadly and effective at short range and in corners: you should definitively develop your reflex to switch to it in such occasion.

**Play in team and collaborate.** At this level, you should definitively use the radio commands (see §2.3) to spread various information to your team. Another specific aspect of the game is the capacity to heal yourself or a friend when bleeding (default key: `Q`). It happens typically when you are shot. The location in which you were hit and the number of times you were hit determines the amount of health you lose each second. The longer you go without healing the more health you loose. To stop yourself from bleeding you need to bandage your wounds. You can do it yourself. Better, you can be healed by a partner to recover up to 40%, while a medic will be able to heal you back up to 80%.

This promotes teamwork so don't hesitate therefore to ask for a medic (`U 3 3`) and rescue damaged team member. In all cases, pay attention to move **in a safe place** to do it.

**Exploit jumping and sprinting techniques.** The faster you move, the faster and further you jump. Sprinting is a good friend of a UrT jumper, its use is necessary for the most impressive jumps and moves. You should bind the sprint command (`+button8`) to a key closed to your directional keys (I bounded it to `E`, as I use `W`, `S`, `A` and `D` for directional keys). Remember that when sprinting, your stamina bar (see Figure 1) decreases. When your stamina is gone you can't sprint and can't jump as high as before. So, you better take notice of its level at all times.

Two basic techniques will make your movements in the map more efficient:

---

[2]Recall that if you do not use up the full magazine before you reload, you will loose the remaining ammunition in your current clip.
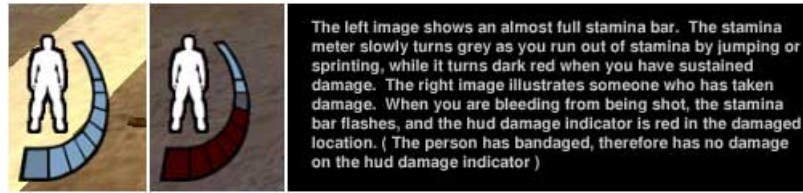
Figure 1: Stamina bar in the HUD of Urban Terror (Source: [4]).

1. *Wall jump*: while in the air after a jump or a fall, you can bounce on a wall (not necessary in front of you) by pressing jump (default: SPACE) when in contact with the wall. You can even repeat this procedure up to 3 times to achieve more complex jump and reaching higher places.

2. *Strafe jump*: this name refers to a common technique in Quake III to reach a higher speed. It consists of changing side after each jump during a movement[3]. For example:

   - first sprint or run in a straight or curved line
   - then jump and and keep turning left + hold left strafe
   - jump and turn right + right strafe
   - jump and turn left + left strafe etc.

   You will probably want to end this sequence by crouching (default key: C): it will make your character sliding to maintain the speed for a short while. You are able to affect the direction of the sliding using the directional key. Another important aspect is that while crouching in this way, you don't consume any more stamina and you're able to focus on a given point of interest.

For more details and techniques, please refer to the Jumping Guide available at `http://www.www0.org/cgi-bin/urtj`.

**Exploit H.E. grenades tempo.** Once unlocked, an HE grenade takes around 4 to 5 seconds to explode. You can play on this time to decide the best moment to throw it away:

- either immediately to protect a path or a place against incoming enemies: hopefully, the grenade will explode while they are reaching the place or prevent them from using it for a short while (Note that this also applied to HK69);
- or after 3 to 4 seconds such that it explodes when reaching the floor such than the ennemy (typically hiden in a corner) won't have time to react and move to a safer place.

Note that it is possible to unlock a grenade and switch to another weapon to cancel the unlocking for further use. This might be useful to avoid

---

[3]Note that in Urban Terror and contrary to Quake III, you can't keep this procedure for long in a single run because of the stamina level.

team kill and a more effective use of them even if I personally consider this as a bug.

**Exploit smoke grenades.** They can cover your movements and be particularly effective when used in combination with tactical goggles: you will be able to locate incoming enemies and shoot with first strike.

## 2.3 Advanced configuration

Apart from the basic key bindings you can operate from the menu SETUP → CONTROLS inside the game, the most useful aspect of UrT is the capacity to fully customize the commands and the bindings use throughout the game to reflect your own style of playing. This is for instance very useful for the radio commands.

All the configuration resided in the configuration file q3config.cfg, located in the q3ut4/ directory of your installed game, typically:

C:\Program Files\UrbanTerror\q3ut4\

~/.q3a/q3ut4/ (make sure to show hidden directories to find it)

~/Library/Application Support/Quake3/q3ut4/

My own configuration file is available at the following url:

http://varrette.gforge.uni.lu/q3ut4/q3config.cfg

The syntax of this file is inherited from Quake III Arena. I will only insist on the *binding* procedure. The general use of this command is:

$$bind\ [key]\ "[command]"$$

You may configure multiple commands with a single key: just separate each commands with a semicolon (;) so that UrT recognizes where each command starts and stops. Note that you can color the text printed in the console using the escape sequence ^n where n is a number between 1 and 8 according to the following associations:

| | | | |
|---|---|---|---|
| 1 | Red | 2 | Green |
| 3 | Yellow | 4 | Blue |
| 5 | Purple | 6 | Cyan |
| 7 | White (Default) | 8 | Black |

Example: "This text is ^1RED^7 and this one ^8BLACK"

I will detail in this document the binding of 2 basic commands: ut_radio (§2.3.1) and ut_weaptoggle (§**??**).

### 2.3.1 Radio command binding

The `ut_radio` command corresponds to radio messages (use U to access the radio interface during the game - see Figure 2) yet it is important to bind the most useful radio messages on a single key, especially if you want to display a colored message in the console which is particularly pertinent for bomb site. Indeed, while the radio messages refer to them as A and B, they correspond in practice to bomb site red and black so a colored message will help your partner to identify more clearly the location you're speaking about.



Figure 2: Radio interface during the game (first level) (Source: [4]).

The table 1 summarizes the most important messages that justify (for me) a binding and (eventually), a more pertinent message displayed on the console. Several variables can be used in such messages, of greater importance stands:

- `$location`: current position in the map;
- `$crosshair`: position in the map pointed by your crosshair;
- `$health`: current health status.

Those preliminary comments being done, here is how to use the table 1. For each message, select a key to bind (let's say for instance `h` to the radio message asking for a medic) then put the following line in your `q3config.cfg`:

bind h "ut_radio 3 3 I need a medic [^1status: $health^7] @ $location"

Eventually, if you don't want to overwrite the default message, simply use:

bind h "ut_radio 3 3"

You can find the full list of the radio messages in the manual of UrT [4].

### 2.3.2 Weapon switch binding

By default, the keys 1 to 6 are assigned for the weapons from the knives to the bomb. It appeared difficult for me to switch quickly with such a configuration to the sidearm or the knife. That's where the `ur_weaptoggle` command is your friend. The format of this command is the following (replace `x` with the key to assign) :

| Type | Menu | Proposed Message |
|---|---|---|
| Order | 2 5 | `Cover me @ $location` |
| | 2 6 | `Requesting backup @ $location` |
| Questions | 3 2 | `Awaiting orders` |
| | 3 3 | `I need a medic [^1status: $health^7] @ $location` |
| | 4 2 | `Objective status?` |
| | 4 4 | `Where's the enemy?` |
| Response | 1 1 | `Affirmative` |
| | 1 2 | `Negative` |
| | 9 4 | `Sorry about that` |
| | 9 9 | `Thanks` |
| Activity | 5 1 | `Enemy spotted` |
| Bomb mode | 8 1 | `Heading to bombsite ^1RED` |
| | 8 2 | `Heading to bombsite ^8BLACK` |
| | 8 3 | `Enemy at bombsite ^1RED` |
| | 8 4 | `Enemy at bombsite ^8BLACK` |
| | 8 5 | `I have the bomb @ $location` |
| | 8 6 | `The bomb is loose!` |

Table 1: Most useful radio message in Urban Terror

```
bind x "ut_weaptoggle [argument]"
bind x "ut_weaptoggle [argument] [argument]"
```

where `argument` can be one of the following: `primary`, `secondary`, `sidearm`, `grenade`, `bomb` or `knife`.

The two argument version permits with the same key to switch from one class of weapon to another. For instance, I personally use the following configuration to easily switch from the primary weapon to the sidearm (useful when in a battle where the magazine becomes empty) or the knife (for a corporal combat):

```
bind f "ut_weaptoggle sidearm primary"
bind q "ut_weaptoggle knife primary"
```

### 2.3.3   Additional sources to tweak your configuration

I just gave some basic elements of customization. You can go further using the following web site:

- `http://q3ut3.free.fr/gear/` (UrbanTerror Gear script Generator) permits the binding of a single key to cycle between several configurations of weapons/equipment which can be useful at the beginning of a new map (for instance to quickly move from a sniper mod with SR8 to AK).

- `http://ucguides.savagehelp.com/` provides a guide for Quake III where some information can be reuse for Urban Terror.

## 2.4   New maps

The best way to install new maps is to auto-download them from the server you are playing. So normally, you have nothing to do (assuming the server is

configured correctly through the `sv_dlURL` directive – see §3.1).

Yet, you may want to install manually new maps. Apart from the basic maps, you can find community created levels on the following website (ordered by personal preferences):

- `http://www.snipersgaulois.com/downloads.php`
- `http://sex-e.clanservers.com/Downloads/c=1.html`
- `http://urt.unfoog.de/q3ut4/`

Just put them into your `q3ut4/` directory.

Those sites propose in general a huge number of maps, some buggy and/or unfinished. If you're only interested in pre-tested maps, you can take a look at my own repository available at: `http://varrette.gforge.uni.lu/q3ut4/`

# 3 Linux server installation and configuration

## 3.1 Basic installation

This section describes the installation of a server under a Linux Debian machine. First proceed to the Linux installation as stated in §2. Then create a dedicated user (`urbanterror`) for running the server, attached to the group `urt`:

```
[root@host]# adduser −−system −−ingroup urt urbanterror
```

The next step is to make this user owning the files coming with Urban Terror, and creating the directory storing the logs and the pid files:

```
[root@host]# chown −R urbanterror:urt /usr/local/games/urbanterror
[root@host]# mkdir /var/log/urbanterror /var/run/urbanterror
[root@host]# chown urbanterror:urt /var/log/urbanterror /var/run/urbanterror
[root@host]# ln −s ˜urbanterror/.q3a/q3ut4/urt.log /var/log/urbanterror/urt.log
```

As the server will be launched under the rights of the `urbanterror` user, the log files of the server will be located in `˜urbanterror/.q3a/q3ut4/urt.log`. Le last instruction creates a symbolic link for the future log file of the server in a more convenient place.

You now have to decide about which server executable you want to use (either 32 bits `ioUrTded.i386` or 64 bits `ioUrTded.x86_64`), make it executable and create the link to it. It follows that for running a 32 bits executable, you will typically run the following commands:

```
[root@host]# cd /usr/local/games/urbanterror
[root@host]# chmod +x ioUrTded.i386
[root@host]# ln −s ioUrTded.i386 urbanterror.server
```

If you followed exactly this tutorial, the configuration file of the server resides in `/usr/local/games/urbanterror/q3ut4/server.cfg`. It is nicely commented so you shouldn't have trouble to adapt it to reflect your own configuration. Yet here are the variables you should pay attention:

```
sets ” Admin” ”yourname”
sets ” Email” ”xxx@xxx.xxx”
```

```
set sv_hostname "your server name, by xxx"
set g_motd "Your stats on xxx" //Your message of the day here,
set sv_joinmessage "Welcome to Urban Terror 4.1" //Your joinmessage here

sets sv_dlURL "urt.unfoog.de" // use this server instead of teh default one

set g_gametype "8" // default to bomb mode ;)

set rconpassword "xxx" //Password to control the server remotely using rcon.

set sv_master2 ""    // leave those empty to prevent notifications of master
set sv_master3 ""    // servers about your server.
set sv_master4 ""

set g_log "urt.log" // name of the log file
//*** B3 Specific settings ***
set g_logSync "3"      //XLR: Unbuffered/direct logging for B3
set sv_zombietime "6" //XLR: Larger zombietime to reduce slot/client mixups for B3
set g_loghits "1" //XLR: headshotcounter and XLRstats bodypart information for B3
```

The `server.cfg` I'm currently using for my own server is proposed in appendix A.1, page 25

Now you should be able to start the server. On Debian, you should be addicted to have a script for startup in the directory `/etc/init.d/`. It appeared difficult to find one so I made one proposed in appendix A.2 page 27. Put this script in the folder `/etc/init.d/`, make it executable (`chmod +x /etc/init.d/urbanterror`). You can now start the server by issuing the command:

[**root@host**]# /etc/init.d/urbanterror start

Now your server is running and should be seen from the client software of your network. Check that you can play on the server, vote to cycle, change map etc. Once ensured that the server is working fine, you can continue to section 4 to configure Big Brother Bot and make your server even more powerful.

## 3.2   Installing new maps and adapting the `mapcycle.txt` file

To install a new map on your server, you have several steps to do:

1. copy the map file (typically `ut4_mapname.pk3`) into the folder
   `/usr/local/games/urbanterror/q3ut4/`;

2. ensure the map file is available for auto-download on the server you configured with the directive `sv_dlURL` in `server.cfg` (see §3.1);

3. adapt your `mapcycle.txt` file to include your new map in the map list (see my script `createmapcycles.sh` below);

4. change the `map` directive at the end of your `server.cfg` to `ut4_mapname` (remove the extension) such that n restart, the new map will be proposed directly;

5. restart the server by issuing the following command:

> [**root@host**]# /etc/init.d/urbanterror restart

To automatically handle new maps and randomizing the maps in the `mapcycle.txt` file, I wrote two scripts:

- `createmapcycles.sh`, to be put together with the file `mapcycle.txt.orig` in the directory `/usr/local/urbanterror/q3ut4/` (both proposed in appendix A.3 page 31).
  This script is responsible for creating a file named `mapcycle.txt.all` listing the default maps (taken from `mapcycle.txt.orig`) and all newly added maps (detecting by a simple `ls` command on the files in the directory with format `ut4_*`).

- `randomize_mapcycle.pl` that randomizes the entries of a file listing the maps to be put in the final `mapcycle.txt`. This Perl script should be placed in the directory `/usr/local/urbanterror/q3ut4/`.

The first script (`createmapcycles.sh`) is typically launched once for each newly installed map, *i.e.* between the steps 4 and 5 described at the beginning of this section.

On the contrary, the second script is typically launched every time the server is started. If you want to activate this feature, simply uncomment the line defining the variable `MAPCYCLE_RANDOMIZE_SCRIPT` in the startup script proposed in appendix A.2.

# 4 Big Brother Bot installation and configuration

Big Brother Bot B3 [2] is a complete and total server administration package for online games. B3 is based on the analysis of the log files populated by a running server. It means that as soon as B3 is able to parse and interpret the logs of a given game, this game is supported. This is the case for Urban Terror. The main interest of B3 resides in the fact that the administrations tasks are handled by plugins written by the community yet with a common interface and installation procedure. This makes B3 very flexible and easy to use.

I detail here the configuration of B3 on your Linux server, assuming you followed the instructions proposed in the previous section.

Note that your main source of information should be the official guidelines provided at the following url:
`http://wiki.github.com/BigBrotherBot/big-brother-bot/installation`
If you experience any problem, don't hesitate to use B3 forum available at
`http://www.bigbrotherbot.com/forums`

## 4.1 Prerequisites

Several components are required before installing B3 in itself:

- A MySQL server and therefore PhPMyAdmin and a Web server (the later being mandatory for `xlrstats`). Under Debian, just run:

```
[root@host]# apt−get install apache2 mysql−server phpmyadmin
[root@host]# ln −s /usr/share/phpmyadmin/ /var/www/phpmyadmin
```

For the configuration, see below.

- some Python stuff (`http://www.python.org`), more particularly: Python 2.3+, Elementtree, Python-MySQL and Python Setuptools. Under Debian, simply run:

```
[root@host]# apt−get install python2.5 python−elementtree \
              python−mysqldb python−setuptools
```

### 4.1.1   Web server configuration

As you plan to use phpMyAdmin, it is mandatory to activate SSL support for the Web server, which means you will have to create a certificate for it. This requires `OpenSSL` so install it:

```
[root@host]# apt−get install openssl
```

Then get the fully qualified host name of the server (run `hostname -f` typically), or use the future url of your server. In the sequel, I'll assume the server to be reached by the name `myurtserver.domain.com`. To create the certificates (valid for 1414 days), run:

```
[root@host]# cd /etc/apache2/
[root@host]# mkdir ssl.key ssl.crt ssl.crl
[root@host]# chmod 700 ssl.key
[root@host]# openssl req −new −x509 −days 1414 −nodes \
   −keyout /etc/apache2/ssl.key/server.key −out /etc/apache2/ssl.crt/server.crt \
   −subj '/O=COM/OU=DOMAIN/CN=myurtserver.domain.com'
```

I don't know how familiar you are with SSL certification, yet you should adapt the subject of the certificate (precised with `-subj`) to reflect your own structure (for the `Organization` name (`COM` here) and your `Organization Unit` name (`DOMAIN` here). The critical point is to place your server fully qualified name as `Common Name`. For more information, read the `README` file coming with apache2:

```
[root@host]# zless /usr/share/doc/apache2.2−common/README.Debian.gz
```

This creates a self-signed certificate `/etc/apache2/ssl.crt/server.crt` and the associated private key `/etc/apache2/ssl.crt/server.key`. Now you should notify the SSL module of Apache about the location of those certificates. Edit `/etc/apache2/mods-available/ssl.conf` and add/update the properties as follows:

```
<IfModule mod_ssl.c>
  ...
  # The certificates of the server to use
  SSLCertificateFile      /etc/apache2/ssl.crt/server.crt
  SSLCertificateKeyFile   /etc/apache2/ssl.key/server.key
  ...
</IfModule>
```

Now comment the `NameVirtualHost` directive in `/etc/apache2/ports.conf`. Then activate the SSL module (together with the rewrite one):

```
[root@host]# a2enmod ssl
[root@host]# a2enmod rewrite
```

You should now update the file configuring the web server by editing the file
`/etc/apache2/site-available/default` as follows:

Listing 1: The `/etc/apache2/site-available/default` configuration file for
your Apache server

```
# To be placed and adapted in /etc/apache2/site−available/default

NameVirtualHost xx.yy.zz.tt:80
NameVirtualHost xx.yy.zz.tt:443

ServerAdmin Your.Mail@domain.com
ServerName myurtserver.domain.com

<Directory />
        Options FollowSymLinks
        AllowOverride None
</Directory>

<Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order Deny,Allow
        Deny from all
        Allow from your.ip.adress.here
        # This directive allows us to have apache2's default start page
        # in /apache2−default/, but still have / go to the right place
        #RedirectMatch ^/$ /apache2−default/
</Directory>

<Directory "/usr/lib/cgi−bin">
        AllowOverride None
        Options +ExecCGI −MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
</Directory>

<Directory /var/www/phpmyadmin/>
        # Restrict phpmyadmin access to just my workstation
        Order Deny,Allow
        Deny from all
        Allow from your.ip.adress.here
</Directory>

<VirtualHost xx.yy.zz.tt:80>
        SSLEngine Off
        # Automatically redirect to https
        RewriteEngine on
        RewriteRule ^/(.*)$ https://%{SERVER_NAME}/$1 [R=permanent]

        ErrorLog /var/log/apache2/error.log
        # Possible values include: debug, info, notice, warn, error, crit, alert, emerg.
        LogLevel warn
        CustomLog /var/log/apache2/access.log combined
        ServerSignature Off
</VirtualHost>

<VirtualHost xx.yy.zz.tt:443>
        SSLEngine On
        DocumentRoot /var/www/
        #### PhpMyAdmin ####
        Alias /phpmyadmin     /var/www/phpmyadmin/
```

```
        ErrorLog /var/log/apache2/error.log
        # Possible values include: debug, info, notice, warn, error, crit, alert, emerg.
        LogLevel warn
        CustomLog /var/log/apache2/access.log combined
        ServerSignature Off
</VirtualHost>
```

Now start the web server by running:

```
[root@host]# apache2ctl restart
```

(You can also use "/etc/init.d/apache2 restart"). Check that the server works by opening a browser in the url http://myurtserver.domain.com. You should be automatically redirect to the url https://myurtserver.domain.com.

### 4.1.2  MySQL configuration

Assuming the MySQL server to be running (/etc/init.d/mysql restart eventually), you should be able also to use PHPMyAdmin by accessing to the url https://myurtserver.domain.com/phpmyadmin/. Note that the server is configured the authorize the access to this interface only for the machine with the IP address your.ip.adress.here according to configuration of the Apache server (see §4.1.1). In case of trouble, your very first reflex (as always) is to check the problem in the logs. To check them online, run in a separate terminal (CTRL-C to quit):

```
[root@host]# tail −f /var/log/apache2/error.log
```

In all case, once you're able to interact with the MySQL server, your very first task is to setup a root password for it (blank by default). Prefer using PH-PMyAdmin for that (go to the Privilege section) or run the following command (just ensure you flush by after the history of the commands to avoid password retrieval – this is not perfect but a minimum security measure):

```
[root@host]# mysqladmin −u root password "your_password_here"
[root@host]# history −c
```

Using PHPMyAdmin, create a MySQL user b3 (generate a random password for him and keep it in a safe place such as an encrypted file) with full right on a database named b3. Create also a MySQL user b3_ro with read-only access to the database b3 (*i.e.* with privileges limited to the SELECT command). Again, generate a random password for him and keep it in a safe place. In the sequel, I'll assume those users to be configured as follows:

- MySQL user b3: password __b3_password__;
- MySQL user b3_ro: password __b3_ro_password__.

## 4.2  B3 installation

You are now ready to install B3 (When writing this tutorial, the latest release of B3 was the version 1.1.4b). Simply run:

```
[root@host]# easy_install −U b3
```

*Note:* this will install b3 in the following directory:
`/usr/lib/python2.5/site-packages/b3-1.1.4b-py2.5.egg/b3`
In the sequel and to make things simpler (and shorter), I will refer to this directory as `$B3_EGG_INSTALLDIR`.

### 4.2.1 Preparing the database

Retrieve the file `$B3_EGG_INSTALLDIR/doc/b3.sql` from the server (typically using `scp`). Then, use PHPMyAdmin to connect to the database `b3` (created during the process described in §4.1.2) as user `b3`. Select the menu `Import`, then choose the previously retrieved file, `b3.sql` and finally click on `Go`. This should populate the database `b3` with the tables required by Big Brother Bot.

### 4.2.2 B3 Configuration

Create the directory structure for hosting the B3 configuration as follows:

```
[root@host]# mkdir ˜urbanterror/b3
[root@host]# cd ˜urbanterror/b3
[root@host]# mkdir −p conf extplugins/conf init.d plugins_toinstall
[root@host]# cp $B3_EGG_INSTALLDIR/conf/∗ ˜urbanterror/b3/conf/
[root@host]# ln −s /var/log/urbanterror/b3.log b3.log
[root@host]# chown −R urbanterror:urt ˜urbanterror/b3
```

The main configuration file for B3 is named `b3.xml` and is located in the directory `/home/urbanterror/b3/conf/`. Edit and adapt it as the one proposed in appendix A.5 page 35. In particular, it is very important to disable Punk-Buster otherwise you will get the error message "*[pm] Please try your command after you have been authenticated*" when playing on the server and trying to register.

The configuration proposed in appendix A.5 activate a single plugin called `admin`. This is for a first check and we will see later how to add and activate new plugins.

The only step missing for the installation is a startup script to manage B3 `/etc/init.d/bigbrotherbot`. Mine is proposed in appendix A.6 page 36. Assuming your UrT server is running (*i.e.* the log files are populated), you should be able to launch b3 as follows:

```
[root@host]# /etc/init.d/bigbrotherbot start
```

To check this is working, connect to the server and once in the game, type as a message `!register`. This will register yourself in B3 (assuming your player name is `Toto`, you should receive as an answer in the console the fact that `Toto` has been put in the `User` group. In all case, detect the anomalies by monitoring the log files of b3 and running in a separate terminal:

```
[root@host]# tail −f /var/log/urbanterror/b3.log
```

You shall now familiarize yourself with the commands available under B3 with the plugin `admin`. For this, you shall refer to the official manual available at: `http://wiki.github.com/BigBrotherBot/big-brother-bot/manual-commands`

You will notice that each command comes with a level to authorize or not the access to the command. To make yourself admin of B3 (*i.e.* able to launch all commands), the best is to update the table `clients` in the `b3` database. Locate the entry with your player name (field `name`). The value associated to your name in the `group_bits` fields should be 1. This translate the fact that you have been put in the User group. Change this value to 128 to make you an admin of B3. You might want to delegate some administration tasks to other players. For this, make their `group_bits` to 20 typically.

## 4.3   Configuring the default plugins

B3 comes with a few plugins more or less relevant for Urban Terror summarized in Table 2.

| | |
|---|---|
| admin | Provides the majority of B3 functionality such as kicking, banning, user management, and warnings |
| adv | Advertise messages periodically |
| censor | Warn users when using banned words |
| stats | Simple transient stats that track stats while the users is connected |
| pingwatch | Warn users for high/low pings and kick when the ping goes to high |
| spamcontrol | Warn users for saying too many messages in a short period of time |
| status | Dumps an XML file to the server periodically with B3 and user information to be used on your website to display game status |
| tk | Monitors team damage and team kills |
| welcome | Welcome messages for new and returning users |

Table 2: Default plugins proposed with Big Brother Bot

Until now, B3 is configured with the `admin` plugin enabled. Before going further, a few elements should be known about the plugins for B3:

- A given plugin `toto` is written in Python in a file named `toto.py`. This file is located in `$B3_EGG_INSTALLDIR/plugins/` (this holds for the default plugins). On startup, B3 eventually compile this plugin to generate a new file `toto.pyc` for a more efficient execution. Therefore if for any reason you decide to modify the source file of a plugin, remember to delete the compiled file before restarting B3. Otherwise your modification won't be taken into account. You may claim you will nether modify plugins sources files, yet it may happens if you want to change the position in the screen where a given message is displayed. You can alter this location as follows:

    - Upper left (server announce area): `self.console.write('blabla')` or `self.console.broadcast('blabla')`

    - Lower left (chat area): `self.console.say('blabla')`

    - Center screen (Big text): `self.console.write('bigtext blabla')`

- A given plugin `toto` comes with a configuration file `plugins_toto.xml` that respects the following format:

```
<configuration plugin="toto">
  ...
</configuration>
```

This configuration files is located by default in `$B3_EGG_INSTALLDIR/conf/` yet remember that I preferred to put the default configurations files in `/home/urbanterror/b3/conf/`. Actually, the exact location of plugins configuration files is specified by the `config` attribute of the `plugin` anchor used in the configuration file `b3.xml`.

- External plugins are handled in a separate directory, either in the default location `$B3_EGG_INSTALLDIR/extplugins/` or, in the hierarchy proposed in this document, in `/home/urbanterror/b3/extplugins/`. This directory will contain the Python sources files of the plugins together with a sub-directory `conf/` which host the plugins configuration files.

- Activating a plugin is done through the configuation file `b3.xml` in the `plugins` section. Each plugin is characterized by a name, a unique priority value (within the list of your plugins) and a path to the configuration file as follows:

```
<plugins>
   <plugin name="plugin_name" priority="n" config="/path/to/plugin_conf.xml"/>
</plugins>
```

  The name provided should correspond to the basename of the Python source file (without the extension `.py`). B3 will search for such a file in the default installation directory (*i.e.* `$B3_EGG_INSTALLDIR/`), then in the default external plugin directory (*i.e.* `$B3_EGG_INSTALLDIR/extplugins/`) and finally in a user-defined external directory specified in a `settings` section of `b3.xml`.

  For instance, in the configuration files proposed in appendix A.5, we setup this external directory to `/home/urbanterror/b3/extplugins` as follows:

```
<settings name="plugins">
   <set name="external_dir">/home/urbanterror/b3/extplugins</set>
</settings>
```

  Note that when specifying the path to the plugin configuration file, you can use the sequence `@b3/` to refer to `$B3_EGG_INSTALLDIR/`.

You can now activate one of the default plugins summarized in the table 2. For instance to activate the plugin `welcome`:

1. Examine `/home/urbanterror/b3/conf/plugin_welcome.xml` to eventually adapt the configuration to suits your needs;

2. Select a priority value $n$ (3 in the sequel) and add a new `plugin` anchor in the `plugins` section of `b3.xml`:

```
<plugins>
   ...
   <plugin name="welcome" priority="3"
           config="/home/urbanterror/b3/conf/plugin_welcome.xml"/>
</plugins>
```

3. Restart B3:

`[`**`root@host`**`]# /etc/init.d/bigbrotherbot restart`

## 4.4   General procedure to install third party plugins

The flexibility of B3 permits the availability of various third party plugins. You will find them in the `Plugins/Releases` section of the Big Brother Bot forum (indeed a very nice source of info for you):

`http://www.bigbrotherbot.com/forums/index.php?board=17.0`

On this page, you should find a list of released plugins, each of them coming with a short description, a download location and an install guide. You shall always refer to the install guide yet here is the general procedure to install and configure a third plugin. To make this procedure concrete, I'll illustrate it on the `heashotsurt` plugin (version `0.2.0`) which counts the number of headshot kills made by the players:

1. Download the plugin sources `heashotsurt-v0.2.0.zip` and unzip this file into the folder:
   `/home/urbanterror/b3/plugins_toinstall/heashotsurt-v0.2.0`

2. Copy (or make a symbolic link of) the Python source file `heashotsurt.py` into the external plugin directory `/home/urbanterror/b3/extplugins/`

3. Copy (or make a symbolic link of) the configuration file `headshots.xml` into the external plugin configuration directory
   `/home/urbanterror/b3/extplugins/conf/`

4. As for default plugins, select a unique priority value $n$ (14 for instance) and add a new `plugin` anchor in the `plugins` section of `b3.xml`:

   ```
   <plugin name="headshotsurt" priority="14"
           config="/home/urbanterror/b3/extplugins/conf/headshots.xml"/>
   ```

5. Restart B3:

   `[`**`root@host`**`]# /etc/init.d/bigbrotherbot restart`

This procedure holds for most of the external plugins. I personally installed in this way the following plugins:

- `PowerAdminUrt` (version `1.4.0b3`)
  `http://www.bigbrotherbot.com/forums/index.php?topic=426.0`
  Remember to edit the file `extplugins/conf/poweradminurt.xml` and disable the `rotationmanager` by putting the value False as follows:

  ```
  <settings name="rotationmanager">
      <!-- enable the rotationmanager? -->
      <set name="rm_enable">False</set>
      ...
  </settings>
  ```

- `headshotsurt` (version 0.2.0)
  http://www.bigbrotherbot.com/forums/index.php?topic=367.0
- `topkiller` (version 1.0.4)
  http://www.bigbrotherbot.com/forums/index.php?topic=574.0
- `Spree` (version 1.1)
  http://www.bigbrotherbot.com/forums/index.php?topic=864.0

I now dedicated a full section to one of the most interesting external plugin which comes with B3: `xlrstats`

## 4.5 Installing XLRstats

XLRstats (http://www.xlrstats.com/) is a statistics plugin for BigBrother-Bot (B3) and it stores all kill-events in the MySQL database used by B3. Stats are available in real time in the game using the `!xlrstats` command in chat, but much more can be viewed in the XLRstats webfront (such as weapon usage, ranks, medals etc.). The sources of the plugin are part of B3 installation (we won't touch them). You still need to download the following elements (put them into `~urbanterror/b3/plugins_toinstall/`):

- The latest webfront `xlrstats-web-v2` (version 2.0.6), available on
  http://github.com/xlr8or/xlrstats-web-v2/tree;
- The image pack for UrT (`xlrstats-imagepack-urt`), to get from
  http://xlr8or.snt.utwente.nl/forum/index.php?action=downloads.

You first need to create the extra tables in the `b3` database. The procedure is similar to the one explained in §4.2.1, this time using the file present in the sub-directory `xlrstats-web-v2/xlrstats/sql/xlrstats.sql`
Once this is done, two B3 plugins should be installed:

1. `status`, one of the default plugins (see §4.3).
   Just edit `/home/urbanterror/b3/conf/plugin_status.xml` as follows:

   ```
   <configuration plugin="status">
           <settings name="settings">
                   <set name="interval">60</set>
                   <set name="output_file">/var/www/status.xml</set>
           </settings>
   </configuration>
   ```

   This will configure the plugin to populate the file `/var/www/status.xml` for which the web server should have read access. In this purpose, run the following commands:

   ```
   [root@host]# touch /var/www/status.xml
   [root@host]# chown urbanterror:www−data /var/www/status.xml
   [root@host]# chmod 644 /var/www/status.xml
   ```

2. the `ctime` plugin that comes with XLRStats webfront in the zip file `b3-plugins/ctime.zip`. Once unpacked, create the extra tables in the `b3` database using the `ctime.sql` file (re-use the procedure proposed in §4.2.1). Then configure the plugin as explained in §4.4.

As always, the plugins are enabled in B3 as soon as you add them in the `plugins` section of `b3.xml` (just adapt the priority values eventually):

```
<plugins>
    ...
    <plugin name="xlrstats" priority="2" config="@b3/extplugins/conf/xlrstats.xml"/>
    ...
    <plugin name="status" priority="10"
            config="/home/urbanterror/b3/conf/plugin_status.xml"/>
    <plugin name="ctime" priority="11"
            config="/home/urbanterror/b3/extplugins/conf/plugin_ctime.xml"/>
    ...
</plugins>
```

Restart B3 and ensure everything is working correctly (check the log file `/var/log/urbanterror/b3.log` eventually). Now we can setup the webfront (take a look at `http://www.xlrstats.com/` for the reference webfront documentation). First copy the files of the webfront into a more convenient place:

```
[root@host]# cd /home/urbanterror/b3/plugins_toinstall/xlrstats−web−v2/xlrstats
[root@host]# cp −r xlrstats /usr/share/
[root@host]# cd ../../
[root@host]# cp −r xlrstats−imagepack−urt/images/∗ /usr/share/xlrstats/images/
[root@host]# chown −R urbanterror:www−data /usr/share/
[root@host]# chmod −R g+r /usr/share/xlrstats
[root@host]# ln −s /usr/share/xlrstats/ /var/www/xlrstats
```

Then make the folders `dynamic` and `config` writable by the webserver:

```
[root@host]# chmod 770 /var/www/xlrstats/dynamic
[root@host]# chmod 770 /var/www/xlrstats/config
```

Update `/etc/apache2/sites-available/default` to include the `xlrstats` directory (typically to limit access to it to a few IP addresses):

```
<Directory /var/www/xlrstats/>
        Order Deny,Allow
        Deny from all
        Allow from 127.0.0.1
        Allow from toto.ip.address
        Allow from tata.ip.address
</Directory>

<VirtualHost xx.yy.zz.tt:443>
        ...
        Alias /stats   /var/www/xlrstats
        ...
</VirtualHost>
```

Finally restart the web server:

```
[root@host]# apache2ctl restart
```

Now go to the url `http://myurtserver.domain.com/stats/install/` Select ”*Start Install*”, then ”*Next*”. In the ”MySQL Connection Settings” panel, enter the following configuration (adapt the password):

- Host name: `localhost`

- MySQL Username : `b3_ro`
- MySQL Password: `__b3_ro_password__`
- MySQL Database Name: `b3`

In the "Basic Game & Server Settings", make sure you select the game Urban Terror, enter the public IP of the server (`xx.yy.zz.tt:27960` – see §4.1.1). The B3 status url is `http://myurtserver.domain.com/status.xml`

Once everything is done, a file `config/statsconfig.php` is created. Ensure this file has the permissions `750`. You should now go to the url `http://myurtserver.domain.com/stats/config/install_award_idents.php` to setup the medals etc. (actually, you should do it every time you experience a trouble with awards). You can now delete the `install` directory and enjoy XLRStats at `http://myurtserver.domain.com/stats/`.

# A Appendix

**Important Note**: the scripts provided in this appendix are available for download at the following url:

> `http://varrette.gforge.uni.lu/q3ut4/../config_files/`

## A.1 My personal `server.cfg`

Listing 2: The `/usr/local/urbanterror/q3ut4/server.cfg` configuration file

```
//*** Administrator Info, shows in some gamebrowsers ***
sets " Admin" "Falkor" //Uses a space in front so it shows up at the top of the properties  list
sets " Email" "falkor@xxx.xx"

//*** Server Name and Daily Message ***
set sv_hostname "TeamBouchry UrT Linux Server, by Falkor" //Your servername here
set g_motd "Your stats on https://falkordesktop.uni.lux/stats/" //Your message of the day here,
                                                    // displayed while connecting
set sv_joinmessage "Welcome to TeamBouchry" //Your joinmessage here, it is displayed when
                                  //the game is joined

//*** General Game Settings ***
set sv_maxclients "12" //max clientslots  available  on the server
set g_maxGameClients "0" //max clients that can actually join the game.
set sv_privateClients "0" //Amount of private slots.
set g_gametype "8" //0=FreeForAll, 3=TeamDeathMatch, 4=Team Survivor, 5=Follow the Leader,
                //6=Capture and Hold, 7=Capture The Flag, 8=Bombmode
set sv_dlURL  " varrette . gforge.uni.lu"
#sets sv_dlURL "urt.unfoog.de"
//Sets the address for auto−downloading. Auto−download only works on
//ioUrbanTerror−clients, not quake3−clients. The client will try to download
//<sv_dlURL>/q3ut4/mapname.pk3. So if your server is running ut4_coolmap and
//sv_dlURL is set to 'yoursite.com/maps', make sure the maps is hosted at
//http://www.yoursite.com/maps/q3ut4/ut4_coolmap.pk3. Leaving this set
//'urbanterror.net' will make it use a map mirror with the most common maps on
//it. If you got your own hosting, please us that though, to save bandwith.

//*** Passwords ***
set rconpassword "yourpasswordhere" //Password to control the server remotely using rcon.
set sv_privatePassword "" //password for private  slots
set g_password "" //password for the server. Nothing = public

//*** Limits/times ***
set timelimit "14" //time in minutes before map is over, 0=never
set fraglimit "20" //amount of points to be scored  before  map is over, 0=never
set capturelimit "9" //amount of flagcaps before map is over, 0=never
set g_warmup "15" //time in seconds before game starts when changed to a new map.
                // Gives slower computers time to load before game starts

//*** Respawning *** (FFA, TDM, CAH, CTF)
set g_respawnDelay "8" //seconds before respawn, ignored when g_waverespawns is 1
set g_forcerespawn "20" //seconds before respawn is forced, even when player did not press fire
set g_waverespawns "0" //use waverespawns, meaning everybody in team respawns at the same time
set g_bluewave "15" //seconds between blue waverespawns, ignored when g_waverespawns is 0
set g_redwave "15" //seconds between red waverespawns, ignored when g_waverespawns is 0
set g_respawnProtection "2" //amount of seconds a spawning players is protected from damage

//*** Rules ***
set g_deadchat "2" //Determines if alive players can see dead players message.
                // 0=living players can not see dead players chat
                // 1=living players see only team−messages from dead teammembers
                // 2=living players also see normal chats from dead players
set g_antiwarp "1" //enable or disable  antiwarp.
```

```
set g_antiwarptol "50" //tolerance of the antiwarp. Higher = more tolerant. 50=default
set g_gear "0" //bitmask that decides which votes are allowed and which not.
              // Check http://www.urbanterror.net/gear_calc.html to find the correct number
set g_allowvote "541473343" //bitmask that decides which votes are allowed and which not.
                            // Check http://www.urbanterror.net/allowvote_calc.html
set g_failedvotetime "0" //time (s) before someone can call another vote after another has failed
set g_followstrict "1" //1=no haunting of enemies when dead
set sv_floodprotect "0" //1=stops clients from spamming many chatlines


//*** Matchmode ***
set g_matchmode "0" //matchmode is for matchplay. Features timeouts and ready-commands
set g_timeouts "3" //ammount of timeouts that a team can do per map
set g_timeoutlength "240" //length of the timeout
set g_pauselength "0" //length of a pause. This can only be done by rcon. 0=indefinatly


//*** Team Game Settings ***
set g_friendlyFire "1" // 0=no friendlyfire 1=friendlyfire on, kick after too many TK's
                       // 2=friendlyfire on, no kicks
set g_maxteamkills "3" //amount of TK's before you get kicked when friendlyfire is 1
set g_teamkillsforgettime "300" //amount of seconds before TK's are forgotten
set g_teamautojoin "0" //force players to autojoin on connect, instead of letting them spec untill
                       // they join themselves
set g_teamForceBalance "1" //if on, you can't join a team when it has more players then the other
set g_maintainTeam "1" //when switching maps, players will stay in their team
set g_teamnamered "Al Quaida" //name for the red team, nothing = Red Dragons
set g_teamnameblue "SWAT" //name for the blue team, nothing = SWAT
set g_swaproles "0" //When map is over, play it again with the teams swapped
                    // After that, change map.
                    // 0=change map immediatly when map is over, no swapping of teams


//*** Team Survivor/Bombmode/Follow the Leader Specific Settings ***
set g_maxrounds "11" //number of rounds before map is over, 0=never
set g_RoundTime "3" //maximum minutes a round can take
set g_survivorrule "0" //0=teams don't get a point when time is up before everyone is dead.
                       //1=team with most players left gets point
set g_suddendeath "1" //when map is over and both teams have same score, add another round
set g_bombdefusetime "10" //seconds it takes to defuse bomb
set g_bombexplodetime "40" //seconds before bomb goes off after planting


//*** Capture the flag Specific Settings ***
set g_flagreturntime "30" //if a flag is dropped, return it after this amount of seconds
set g_hotpotato "2" //when both flags are taken, they will explode after this amount of minutes


//*** Advanced settings *** Dont change, unless you know what you are doing
set sv_strictauth "0" //1=check for valid cdkey, this means ioUrbanTerror players will
                      // not be able to join
set sv_pure "1" //dont let players load modified pk3-files
set sv_maxRate "0" //maximum traffic per second the server will send per client. 25000 or 0=max
set sv_timeout "180" //time in seconds before player with a interupted connection will be kicked
set g_inactivity "0" //time in seconds before a non-moving player will be kicked


//*** Master Servers *** Servers the server will report to if 'dedicated' is set to 2.
//                       When set to 1, it doesn't report.
set sv_master1 "" //This one will be set automatically by the game-engine, so just leave it blank
set sv_master2 ""
#master.urbanterror.net"
set sv_master3 ""
#master2.urbanterror.net"
set sv_master4 ""
#master.quake3arena.com"
set sv_master5 ""


//*** Other Settings ***
set g_armbands "1" //determines the behaviour of armbandcolor (also on playerlist & minimap).
                   // 0=player's choice, set with cg_rgb
                   // 1=Based on teamcolor (red or blue)
```

```
                    // 2=assigned by server (random)
set sv_maxping "0" //max ping a client may have when connecting to the server
set sv_minping "0" //min ping a client may have when connecting to the server
set g_allowchat "2"  //0= no chatting at all 1=teamchats only 2=all chats
set g_log "urt.log"  //name of the logfile (go into q3ut4 directory) . Empty ("") means no log.
set g_logsync "3" // XLR: Unbuffered/direct logging for B3
set sv_zombietime "6" // XLR: Larger zombietime to reduce slot/client mixups for B3
set g_loghits "1" // XLR: headshotcounter and XLRstats bodypart information for B3
set g_logroll "0"  //create new log every now and then, instead of always using the same one
set logfile "0" //additional logging in seperate qconsole.log file . 1=buffered, 2=synced
set g_cahtime "60" //Interval in seconds of awarding points for flags in C&H gamemode

//*** Map Rotation ***
set g_mapcycle "mapcycle.txt" //name of mapcycle−file, located in q3ut4 directory
map ut4_turnpike


//*** Anti Cheat ***
//pb_sv_enable //to enable PB, remove the // at the beginning of this line
              // (only works when using Quake 3 Arena, not ioUrbanTerror)
set sv_battleye "0" //Keep this disabled, BattlEye is dead
```

## A.2   `/etc/init.d/urbanterror`

Remember to make this script executable (`chmod +x`) once copied in the directory `/etc/init.d/`.

Listing 3: The startup script `/etc/init.d/urbanterror`

```
#! /bin/bash
#
# File    : urbanterror
# Author : Sebastien VARRETTE <Sebastien.Varrette@uni.lu>
#          (Web page : http://varrette.gforge.uni.lu)
# Creation date : 23 Jul 2009
# Time−stamp: <2009−07−28 16:10:39 svarrette>
# $Id$
#
# Copyright (c) 2009 Sebastien Varrette (Sebastien.Varrette@uni.lu)
#
# Description : init.d startup script with LSB support for XXXX
#              LSB support: http://wiki.debian.org/LSBInitScripts
# See http://varrette.gforge.uni.lu/download/scripts/init.d.template_example
#
# This program is free software; you can redistribute it and/or modify it under the terms
# of the GNU General Public License as published by the Free Software Foundation; either
# version 2 of the License, or (at your option) any later version.
# This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
# without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
# See the GNU General Public License for more details.
# You should have received a copy of the GNU General Public License along with this
# program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street,
# Fifth Floor, Boston, MA 02110−1301, USA.
#
# Sebastien Varrette <Sebastien.Varrette@uni.lu>
# University of Luxembourg
# 6, rue Richard Coudenhove−Kalergi
# L−1359 Luxembourg, LUXEMBOURG
#
# Note: inspired by /etc/init.d/skeleton and utrac debian/init.d.lsb.ex file
# see http://bzr.harobed.org/utrac/utrac
#
### BEGIN INIT INFO
# Provides:          urbanterror
# Required−Start:    $local_fs
# Required−Stop:     $local_fs
```

27

```
# Default−Start:     2 3 4 5
# Default−Stop:      0 1 6
# Short−Description: Urban Terror server
# Description:       Startup script for Urban Terror server
### END INIT INFO

VERSION=0.1
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
DESC="Urban Terror server"
NAME=urbanterror

UrT_BASEDIR=/usr/local/games/urbanterror
DAEMON=${UrT_BASEDIR}/${NAME}.server
DAEMON_USER=urbanterror
DAEMON_GROUP=urt
DAEMON_PORT=27960
DAEMON_OPTS="+set fs_game q3ut4 +set dedicated 2 +set net_port ${DAEMON_PORT} \
            +set com_hunkmegs 128 +exec server.cfg"
DAEMON_START_OPTS="−−chdir ${UrT_BASEDIR}"
SCRIPTNAME=/etc/init.d/${NAME}
PIDFILE=/var/run/${NAME}/${NAME}.pid # PID file

# Uncomment the next line to include the randomization of the mapcycle
#MAPCYCLE_RANDOMIZE_SCRIPT="${UrT_BASEDIR}/q3ut4/randomize_mapcycle.pl −−quiet"
# Uncomment the next line to include Big Brother Bot startup in the same time UrT is launched
#B3_SCRIPTNAME=/etc/init.d/bigbrotherbot

# Load the VERBOSE setting and other rcS variables
. /lib/init/vars.sh

# Define LSB log_* functions.
# Depend on lsb−base (>= 3.0−6) to ensure that this file is present.
. /lib/lsb/init−functions

# Include defaults if available
[ −r /etc/default/${NAME} ] && . /etc/default/$NAME

# Exit if the server executable is not present
if [ ! −x "${DAEMON}" ]; then
    echo "Could not find ${DAEMON}"
    exit 99
fi

# Check that the user exists (if a user is set)
if [ −n "$DAEMON_USER" ] ; then
    if [ ! `getent passwd | grep "^${DAEMON_USER}:"` ]; then
        log_failure_msg "The user $DAEMON_USER, required to run ${NAME} does not exist."
        exit 1
    fi
fi
# Check that the group exists (if a group is set)
if [ −n "$DAEMON_GROUP" ] ; then
    if [ ! `getent group | grep "^${DAEMON_GROUP}:"` ]; then
        log_failure_msg "The group $DAEMON_GROUP, required to run ${NAME} does not exist."
        exit 1
    fi
fi

# configure pid file
PIDDIR=`dirname "${PIDFILE}"`
if [ "${PIDDIR}" −a "${PIDDIR}" != "/var/run" ] ; then
    [ ! −d "${PIDDIR}" ] && mkdir "${PIDDIR}"
    [ "${DAEMON_USER}" ] && chown "${DAEMON_USER}" "${PIDDIR}"
    [ "${DAEMON_GROUP}" ] && chgrp "${DAEMON_GROUP}" "${PIDDIR}"
fi
```

```
######### specific functions ##########

# Various print functions
print_error_and_exit () { echo $1; exit 1; }
print_usage() {
    cat <<EOF
This is $SCRIPTNAME version "$VERSION", ${DESC} startup script.

    Copyright (c) 2009 Sebastien Varrette  (http://varrette.gforge.uni.lu)

Usage: $SCRIPTNAME {start|stop|restart|status|restart|reload|force−reload}

This is  free  software; see the source for copying conditions.   There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
EOF
}


###
# check the daemon/service
##
check_running_daemon() {
    local  basedaemon=`basename ${DAEMON}`
    local  nb_process=`ps ax | grep $basedaemon | grep −v grep | wc −l`
    [ ! −f "${PIDFILE}" ] && return 1
    if [ $nb_process −gt 1 ]; then
        log_warning_msg "is running multiple times"
        return 0
    fi
    pidofproc −p ${PIDFILE} $basedaemon >/dev/null
    return $?
}


###
# operate daemon status checking
##
do_status() {
#    status_of_proc  ${DAEMON} ${NAME} −p ${PIDFILE}
#    return $?
    log_daemon_msg "Checking status of $DESC" "$NAME"
    check_running_daemon
    RETVAL=$?
    [ $RETVAL −eq 0 ] && log_progress_msg "is running" || \
        log_progress_msg "is  not running"
    log_end_msg 0
    return $RETVAL
}


###
# start the daemon/service
##
do_start() {
    ${MAPCYCLE_RANDOMIZE_SCRIPT} # randomize the mapcycle
    if [ −z "${DAEMON_USER}" ] ; then
        start_daemon −p ${PIDFILE} ${DAEMON} −− ${DAEMON_OPTS}
        RETVAL=$?
    else
        start−stop−daemon \
            −−start −−quiet −−chuid ${DAEMON_USER}:${DAEMON_GROUP} \
            −−pidfile ${PIDFILE} −−make−pidfile −−background −−oknodo \
            −−startas ${DAEMON} ${DAEMON_START_OPTS} −− ${DAEMON_OPTS}
        RETVAL=$?
    fi
    return $RETVAL
}


###
```

```
# stop the daemon/service
##
do_stop() {
    if [ −z "${DAEMON_USER}" ] ; then
        killproc −p ${PIDFILE} ${DAEMON}
        RETVAL=$?
    else
        start−stop−daemon \
            −−stop −−signal 15 −−retry 5 −−quiet −−oknodo \
            −−pidfile ${PIDFILE} −−user ${DAEMON_USER} −−exec ${DAEMON}
        RETVAL=$?
    fi
    # Many daemons don't delete their pidfiles when they exit.
    [ "$RETVAL" = 0 ] && rm −f $PIDFILE
    return $RETVAL
}


###
# reload the daemon/service
##
do_reload() {
    if check_running_daemon ; then
        start−stop−daemon −−stop −−signal 1 −−quiet −−pidfile ${PIDFILE} −−name ${NAME}
        return $?
    fi
    return 0
}


###
# use the Bib Brother Bot startup script
# usage: call_b3 action
##
call_b3 () {
    if [ −n "${B3_SCRIPTNAME}" ]; then
        [ "$1" != status ] && sleep 4
        ${B3_SCRIPTNAME} $1
    fi
}


############ proceed to the command−line option ##########
RETVAL=0
case "$1" in
    start)
        [ "$VERBOSE" != no ] && log_daemon_msg "Starting $DESC" "$NAME"
        # Check if it's running first
        if check_running_daemon ; then
            log_progress_msg "is already running"
            log_end_msg 0
            exit 0
        fi
        do_start
        RETVAL=$?
        [ "$VERBOSE" != no ] && log_end_msg $RETVAL
        call_b3 start
        ;;
    stop)
        [ "$VERBOSE" != no ] && log_daemon_msg "Stopping $DESC" "$NAME"
        # Check if it's running first
        if check_running_daemon ; then
            do_stop
            RETVAL=$?
            [ "$VERBOSE" != no ] && log_end_msg $RETVAL
        else
            log_progress_msg "is not running"
            log_end_msg 0
```

```
        fi
        call_b3 stop
        ;;
    reload)
        # Use this if the daemon cannot reload
        log_warning_msg "\nReloading ${NAME}: not implemented, as the daemon"
        log_warning_msg "\ncannot re−read the config file (use restart instead)."
        # ... or this if it can
        #log_daemon_msg "Reloading $DESC configuration files" "$NAME"
        #do_reload
        #log_end_msg $?
        ;;
    restart | force−reload)
        [ "$VERBOSE" != no ] && log_daemon_msg "Restarting $DESC" "$NAME"
        if ! check_running_daemon ; then
            [ "$VERBOSE" != no ] && log_warning_msg "... were not running"
            [ "$VERBOSE" != no ] && log_daemon_msg "Now trying to start $DESC" "$NAME"
        else
            do_stop
            RETVAL=$?
            if [ $RETVAL −ne 0 ]; then
                log_progress_msg "stopping "
                log_end_msg 1
                exit $RETVAL
            fi
        fi
        do_start
        RETVAL=$?
        case "$RETVAL" in
            0) log_end_msg 0 ;;
            ∗) log_progress_msg "starting "; log_end_msg 1 ;;
        esac
        do_status
        call_b3 status
        ;;
    status)
        do_status
        RETVAL=$?
        call_b3 status
        ;;
    ∗)
        print_usage
        RETVAL=3;;
esac
exit $RETVAL

:
```

## A.3   Automatic generation of the file `mapcycle.txt.all` containing all maps available on the server

Put all those elements into the **/usr/local/games/urbanterror/q3ut4/** directory.

Listing 4: The file  `mapcycle.txt.orig` listing the default maps available by default

```
ut4_casa
ut4_kingdom
ut4_turnpike
ut4_abbey
ut4_prague
ut4_mandolin
ut4_uptown
```

```
ut4_algiers
ut4_austria
ut4_elgin
ut4_oildepot
ut4_swim
ut4_harbortown
ut4_ramelle
ut4_toxic
ut4_sanc
ut4_riyadh
ut4_ambush
ut4_eagle
ut4_suburbs
ut4_crossing
ut4_subway
ut4_thingley
```

Listing 5: The script `createmapcycles.sh`

```bash
#!/bin/bash

ORIG=mapcycle.txt.orig
NEW=mapcycle.txt.new
ALL=mapcycle.txt.all

# create mapcycle for new maps
echo -n "" > ${NEW}
for f in `ls ut4_*`; do
        echo `basename $f .pk3` >> ${NEW}
done

# create mapcycle for all maps
#echo -n "" > ${ALL}
cat ${ORIG} > ${ALL}
cat ${NEW} >> ${ALL}
```

## A.4 Randomizing the `mapcycle.txt`

As before, the following script should be placed in the server directory
`/usr/local/games/urbanterror/q3ut4/`. It assumes the presence of a file
named by default `mapcycle.txt.ref` that lists the maps you want to randomize
in the output file `mapcycle.txt`. If you want to use all possible maps with the
file generated by my script `createmapcycles.sh` proposed in the §A.3, simply
create a symbolic link on `mapcycle.txt.all` as follows:

```
[root@host]# cd /usr/local/games/urbanterror/q3ut4/
[root@host]# ln -s mapcycle.txt.all mapcycle.txt.ref
[root@host]# ./randomize_mapcycle.pl
=> Mapcycle File   ...........   /usr/local/games/urbanterror/q3ut4/mapcycle.txt
=> Reference Mapcycle File .. /usr/local/games/urbanterror/q3ut4/mapcycle.txt.ref
=> Number of maps  ...........   46
=> Randomizing /usr/local/games/urbanterror/q3ut4/mapcycle.txt.ref to generate
/usr/local/games/urbanterror/q3ut4/mapcycle.txt
```

Listing 6: The script `randomize_mapcycle.pl`

```perl
#!/usr/bin/perl -w

##################################################
# File   : randomize_mapcycle.pl
```

```perl
# Author : Sebastien VARRETTE <Sebastien.Varrette@uni.lu>
# 10 Apr 2009
#
# Description :
#               See the man page for more information.
#
# Copyright (c) 2009 Sebastien VARRETTE (http://www-id.imag.fr/~svarrett/)
#
# This program is free software; you can redistribute  it and/or
# modify it under the terms of the GNU General Public License
# as published by the Free Software Foundation; either version 2
# of the License, or (at your option) any later version.
#
# This program is distributed in the hope that it  will  be  useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
#
# Sebastien VARRETTE                         \n
# <Sebastien.Varrette@uni.lu>                \n
# University of Luxembourg                   \n
# 6, rue Richard Coudenhove-Kalergi          \n
# L-1359 Luxembourg                          \n
###############################################################
use strict ;
use warnings;

# Used packages
use Getopt::Long;              # For command line management (long version)
use Term::ANSIColor;           # To send the ANSI color-change sequences to the user's terminal
use Pod::Usage;
use List::Util ' shuffle ';
#use Data::Dumper;

# Generic variables
my $VERSION = '0.1';           # Script version
my $VERBOSE = 0;               # option variable for verbose mode with default value (false)
my $DEBUG = 0;                 # option variable for debug mode with default value (false)
my $QUIET = 0;                 # By default, display all informations
my $numargs = scalar(@ARGV);  # Number of arguments
my $command = 'basename $0';   # base command
chomp($command);

my $SIMULATION_MODE = 0;       # By default, don't simulate

# Parse command line
my $mapfile    = "/usr/local/games/urbanterror/q3ut4/mapcycle.txt";
my $mapfile_ref = "$mapfile".".ref";

my $getoptRes = GetOptions('mapfile|f=s' => \$mapfile,              # mapcycle file
                           'ref|r=s'     => \$mapfile_ref,          # mapcycle file reference
                           'dry-run|n' => \$SIMULATION_MODE,   # Simulation mode
                           'verbose|v' => \$VERBOSE,           # Verbose mode
                           'quiet|q'   => \$QUIET,             # Quiet mode
                           'debug'     => sub { $DEBUG = 1; $VERBOSE = 1; }, #Debug mode
                           'help|h'    => sub { pod2usage(-exitval => 1,
                                                    -verbose => 2); }, # Show help
                           'version'   => sub { VERSION_MESSAGE(); exit(0); } #Show version
                          );

PRINT_ERROR_THEN_EXIT("Please check the format of the command-line $!")
  unless ($getoptRes);
```

33

```perl
my @maps = ();

info("Mapcycle File    .............   $mapfile\n" );
info("Reference Mapcycle File ...  $mapfile_ref\n" );
open(MAPREF, "<", "$mapfile_ref") || die "cannot read $mapfile_ref: $!\n";
while (<MAPREF>) {
    @maps = (@maps, $_);
}
close(MAPREF);
info("Number of maps   ............   " . scalar(@maps) . "\n");

my @randommaps = shuffle(@maps);

unless ($SIMULATION_MODE) {
    info("Randomizing $mapfile_ref to generate $mapfile\n");
    open(MAPS, ">", "$mapfile") || die "cannot write $mapfile: $!\n";
    foreach my $map (@randommaps) {
        chomp($map);
        print MAPS "$map\n";
    }
    close(MAPS);
}

############ ----- Sub routines ----- ###########

######
# Print information in the following form: '[$2] $1' ($2='=>' if not submitted)
# usage: info(text [, title ])
##
sub info {
    PRINT_ERROR_THEN_EXIT( '[' . (caller(0))[3] . '] missing text argument') unless @_;
    my $prefix = $_[1] ? $_[1]  : '=>';
    print "$prefix $_[0]" unless $QUIET;
}

####
# Print script version
##
sub VERSION_MESSAGE {
    print <<EOF;
This is $command v$VERSION.
Copyright (c) 2009 Sebastien VARRETTE (http://www-id.imag.fr/~svarrett/)
This is  free  software;  see the source for copying conditions.   There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
EOF
}

########## POD documentation ###########
=pod

=head1 NAME

I<randomize_mapcycle.pl>, a nice script in perl to randomize mapcycle.txt

=head1 SYNOPSIS

    ./randomize_mapcycle.pl [options]

=head1 OPTIONS

The following options are  available :

=over 12

=item B<--debug>
```

Debug mode. Display debugging information probably only relevant to me ;)

=item B<−−dry−run −n>

Simulate the operations to show what would have been done and/or transferred but **do** not perform any backend actions.

=item B<−−help −h>

Display a help screen and quit.

=item B<−−mapfile MAP −f MAP>

Set MAP as the mapcycle.txt file to be generated
Default: /usr/**local**/games/urbanterror/q3ut4/mapcycle.txt

=item B<−−quiet>

Quiet mode. Minimize the number of printed messages and don't ask questions.
Very useful for invoking this script in a crontab yet use with caution has all operations will be performed without your interaction.

=item B<−−ref MAP −r MAP>

Set MAP as the reference mapcycle file that serves as input for this script
(the entries of this file will be randomize to generate the final mapcycle.txt
file .
Default: /usr/local/games/urbanterror/q3ut4/mapcycle.txt.ref


=item B<−−verbose −v>

Verbose mode. Display more information

=item B<−−version>

Display the version number then quit.

=back

=head1 BUGS

Please report bugs to Sebastien VARRETTE <Sebastien.Varrette@uni.lu>

=head1 AUTHOR

Sebastien VARRETTE −− L<http://varrette.gforge.uni.lu/>

=head1 COPYRIGHT

This is a free software. There is NO warranty; not even for
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

=cut

## A.5   B3 main configuration files

Listing 7: The B3 configuration file `/home/urbanterror/b3/conf/b3.xml`

```
<configuration>
  <settings name="b3">
    <set name="parser">iourt41</set>
    <!−− /!\ adapt MySQL password for the b3 user −−>
    <set name="database">mysql://b3:__b3_password__@localhost/b3</set>
    <set name="bot_name">b3</set>
```

```
  <set name="bot_prefix">^0(^2b3^0)^7:</set>
  <set name="time_format">%I:%M%p %Z %m/%d/%y</set>
  <set name="time_zone">CST</set>
  <!-- 9 = verbose, 10 = debug, 21 = bot, 22 = console -->
  <set name="log_level">9</set>
  <set name="logfile">/var/log/urbanterror/b3.log</set>
</settings>
<settings name="server">
  <set name="punkbuster">off</set>
  <set name="rcon_password">yourpasswordhere</set><!-- /!\ see server.cfg -->
  <set name="port">27960</set>
  <set name="game_log">/var/log/urbanterror/urt.log</set>
  <set name="public_ip">xx.yy.zz.tt</set><!-- /!\ see server.cfg -->
  <set name="rcon_ip">xx.yy.zz.tt</set><!-- /!\ see server.cfg -->
</settings>
<settings name="messages">
  <set name="kicked_by">%s^7 was kicked by %s^7 %s</set>
  <set name="kicked">%s^7 was kicked %s</set>
  <set name="banned_by">%s^7 was banned by %s^7 %s</set>
  <set name="banned">%s^7 was banned %s</set>
  <set name="temp_banned_by">%s^7 was temp banned by %s^7 for %s^7 %s</set>
  <set name="temp_banned">%s^7 was temp banned for %s^7 %s</set>
  <set name="unbanned_by">%s^7 was un-banned by %s^7 %s</set>
  <set name="unbanned">%s^7 was un-banned %s</set>
</settings>
<settings name="plugins">
  <set name="external_dir">/home/urbanterror/b3/extplugins</set>
</settings>
<plugins>
  <plugin name="admin" priority="1" config="/home/urbanterror/b3/conf/plugin_admin.xml"/>
</plugins>
</configuration>
```

## A.6  `/etc/init.d/bigbrotherbot`

Listing 8: The startup script `/etc/init.d/bigbrotherbot`

```
#! /bin/sh
### BEGIN INIT INFO
# Provides:          BigBrotherBot
# Required-Start:    $local_fs
# Required-Stop:     $local_fs
# Default-Start:     2 3 4 5
# Default-Stop:      S 0 1 6
# Short-Description: BigBrotherBot
# Description:       BigBrotherBot
### END INIT INFO

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
DESC="BigBrotherBot"
NAME=bigbrotherbot
SCRIPTNAME=/etc/init.d/$NAME
DAEMON=`which python`
DAEMON_USER=urbanterror
DAEMON_GROUP=urt

B3_BASEDIR=/usr/bin
B3_CONFIGFILE="/home/urbanterror/b3/conf/b3.xml"

DAEMON_ARGS="$B3_BASEDIR/b3_run -c $B3_CONFIGFILE"

B3_PID_FILE="/var/run/urbanteror/${NAME}.pid"


# Exit if the package is not installed
if [ ! -x "$DAEMON" ]; then
```

```
        {
            echo "Couldn't find $DAEMON"
            exit 99
        }
fi


PIDDIR=`dirname "$B3_PID_FILE"`
if [ "$PIDDIR" -a "$PIDDIR" != "/var/run" ] ; then
    [ ! -d "$PIDDIR" ] && mkdir "$PIDDIR"
    [ "$DAEMON_USER" ] && chown "$DAEMON_USER" "$PIDDIR"
    [ "$DAEMON_GROUP" ] && chgrp "$DAEMON_GROUP" "$PIDDIR"
fi

# Define LSB log_* functions.
# Depend on lsb-base (>= 3.0-6) to ensure that this file is present.
. /lib/lsb/init-functions

#
# Function that check the daemon/service
#
do_status()
{
    NB_PROCESS=`ps ax | grep b3_run | grep "$B3_CONFIGFILE" | grep -v grep | wc -l`
    if [ $NB_PROCESS -gt 1 ]; then
        log_warning_msg "WARNING: $NAME is running multiple times"
        return 0
    elif [ $NB_PROCESS -eq 1 ]; then
        log_success_msg "$NAME is running"
        return 0
    else
        log_success_msg "$NAME is stopped"
        return 1
    fi
}


#
# Function that starts the daemon/service
#
do_start()
{
    # Return
    #   0 if daemon has been started
    #   1 if daemon was already running
    #   2 if daemon could not be started
    start-stop-daemon --start --quiet --chuid $DAEMON_USER:$DAEMON_GROUP \
        --pidfile $B3_PID_FILE --make-pidfile \
        --background --chdir $B3_BASEDIR --startas $DAEMON -- $DAEMON_ARGS
    return $?
}


#
# Function that stops the daemon/service
#
do_stop()
{
    # Return
    #   0 if daemon has been stopped
    #   1 if daemon was already stopped
    #   2 if daemon could not be stopped
    #   other if a failure occurred
    start-stop-daemon --stop --signal 15 --retry 5 --quiet --pidfile $B3_PID_FILE
    RETVAL="$?"
    [ "$RETVAL" = 0 ] && rm -f $B3_PID_FILE
    [ "$RETVAL" = 2 ] && return 2
    return "$RETVAL"
}
```

```
do_kill () {
    while [ 1 ]; do
        PID=`ps hax | grep "b3_run" | grep "$B3_CONFIGFILE" | grep −v grep | cut −d' ' −f1 | head −n1`
        [ "$PID" = "" ] && return
        log_daemon_msg "killing process [$PID]"
        kill −9 $PID
    done
}

case "$1" in
    start )
        log_daemon_msg "Starting $DESC" "$NAME"
        do_start
        case "$?" in
            0|1)  log_end_msg 0 ;;
            2)    log_end_msg 1 ;;
        esac
        do_status
        ;;
    stop)
        log_daemon_msg "Stopping $DESC" "$NAME"
        do_stop
        case "$?" in
            0|1)  log_end_msg 0 ;;
            2) log_end_msg 1 ;;
        esac
        ;;
    status)
        do_status
        ;;
    kill )
        log_daemon_msg "Killing all process b3_run using $B3_CONFIGFILE"
        do_kill
        log_end_msg 0
        ;;
    restart | force−reload)
        log_daemon_msg "Restarting $DESC" "$NAME"
        do_stop
        case "$?" in
            0|1)
                do_start
                case "$?" in
                    0) log_end_msg 0 ;;
                    1) log_end_msg 1 ;;  # Old process is  still  running
                    ∗) log_end_msg 1 ;;  # Failed to  start
                esac
                do_status
                ;;
            ∗)
        # Failed to stop
                log_end_msg 1
                ;;
        esac
        ;;
    ∗)
        echo "Usage: $SCRIPTNAME {start|stop|restart|force−reload}" >&2
        exit 3
        ;;
esac

:
```

# References

[1] Counterstrike, 2000. [Online] `http://store.steampowered.com/app/10/`.

[2] Big Brother Bot (B3), 2009. [Online] `http://bigbrotherbot.com/`.

[3] ioquake3, 2009. [Online] `http://ioquake3.org/`.

[4] Urban terror, 2009. [Online] `http://www.urbanterror.net/`.