



# L-Fire CTF

*Version ~~1.0~~1.1*

Author: Kai Kenner "L-Fire"

Created 1998 and protected as an Unpublished Work Under the U.S. Copyright Act of 1976.

© Copyright 1998-~~1999~~ Kai Kenner. All rights reserved.

---

# Table of Contents

---

<b>1. Introduction</b>	<b>2</b>
<b>2. Getting Started</b>	<b>2</b>
2.1. Archive Contents .....	3
2.2. Quick Start .....	3
2.3. Default L-Fire CTF Configuration .....	3
2.4. Your Server Configuration.....	4
<b>3. Running L-Fire CTF</b>	<b>4</b>
3.1. Message of the Day .....	4
3.2. Highscores .....	4
3.3. Level Start Countdown.....	5
3.4. Filtering Names and Chat Messages .....	5
3.5. Weaponflags.....	6
3.6. Configuring the HUD .....	6
3.7. IP Banning .....	7
3.8. Silencing Players.....	7
3.9. Logging .....	8
3.10. Using Custom Map Loops.....	9
3.11. Using Entity Files .....	9
3.12. Creating Admin Users .....	10
3.13. Restricting Player Pings .....	12
3.14. Bot Prevention.....	12
3.15. Server Maximum Rate .....	13
3.16. Additional Features .....	13
<b>4. Overtime</b>	<b>14</b>
4.1. Rocket Arena Overtime.....	14
4.2. Sudden Death Overtime.....	15
<b>5. Running Matches</b>	<b>15</b>
5.1. Match Admin Elections.....	15
5.2. Setup Stage .....	16
5.3. Countdown .....	16
5.4. Ending the Match .....	16
5.5. Server Feature Overrides.....	16
5.6. Locking the Match.....	17
<b>6. Appendix</b>	<b>17</b>

6.1. Specifying Player Names .....	17
6.2. Client Commands.....	18
6.3. Match Commands.....	20
6.4. Admin Commands .....	21
6.5. Server Commands .....	23
6.6. Server CVARS .....	24
6.7. Configuration Files .....	32
6.8. Using Outlands II.....	33

**Revision History**

Date	Editor	Description
<u>02/11/99</u>	<u>KK</u>	<u>Update for release 1.1</u>
10/19/98	KK	Update for release 1.0
07/25/98	KK	Update for release 0.88
07/06/98	KK	Release 0.87
06/22/98	KK	Initial release with version 0.86

# 1. Introduction

L-Fire CTF is a server-side mod that significantly enhances Id's official Quake2 CTF.

The goal of L-Fire CTF is not to change the fundamental game or balance of Id's CTF. Instead, this mod adds useful and much needed improvements to an already great game. Many of the improvements directly benefit the players, such as match support with countdown, anti-spam, highscores, and team balancing, but there's also a host of features that are tailor-made for server operators, such as IP banning, custom map loops, permissions for admin users, and various logging options.

## Contact

Visit the L-Fire CTF website at <http://www.planetquake.com/LFire> or email to LFire@yyz.com.

## Copyright/Permissions

L-Fire CTF is copyright © 1998-1999, Kai Kenner

L-Fire CTF was created solely for my own use, so it is distributed AS IS.

You may distribute the L-Fire CTF archive so long as all of the files are distributed together.

You may only distribute L-Fire CTF and its supporting files for free.

# 2. Getting Started

By default, L-Fire CTF is set up as a drop-in replacement for Id's standard CTF. Using L-Fire CTF in its default configuration will in no way change the gameplay or "purity" of Id's CTF. Features that modify the standard game, such as quick weapon switching or weapons banning, are off by default so as not to change the fundamental game that you're used to.

## Note:

Throughout this document, filenames and directory names are given in mixed case. On Linux, all files and directories must be in lower case.

## 2.1. Archive Contents

For your L-Fire CTF binary, you should have one of the following files:

<u>gamex86.dll</u>	<u>L-Fire CTF binary (Win32)</u>
<u>gamei386.so</u>	<u>L-Fire CTF binary (Linux)</u>
<u>gamesparc.so</u>	<u>L-Fire CTF binary (Solaris Sparc)</u>

These are the files and directories ~~that~~ you should have in your L-Fire CTF ~~distribution:~~configuration file archive:

<del>ctf/gamex86.dll</del>	<del>L-Fire CTF binary (Win32)</del>	
<del>ctf/gamei386.so</del>	<del>L-Fire CTF binary (Linux)</del>	
<del>ctf/LFireCfg/Ent</del>	Directory for entity files	
<del>ctf/LFireCfg/HighScores</del>	Directory for highscore files	
<del>ctf/LFireCfg/Logs</del>	Directory for log files	
<del>ctf/LFireCfg/TempScores</del>	Directory for temporary score files	
<del>ctf/LFireCfg/Ent/nobfg_q2ctf4.ent</del>	Custom entity file for Outlands	
<del>ctf/LFireCfg/HighScores/NewScores.txt</del>	Empty highscore file template	
<del>ctf/LFireCfg/HighScores/q2ctf1.txt</del>	Empty highscore file for q2ctf1 (Highscore files included for q2ctf1-q2ctf5 and as2m1-as2m7)	
<del>ctf/LFireCfg/Lfire100.txt</del>	Readme file for version 1.0	
<del>ctf/LFireCfg/LFireCTFManuall100.pdf</del>	<u>LFireCfg/LFireCTFManuall_1.pdf</u>	This manual
<del>ctf/LFireCfg/Admin.cfg</del>	File for admin permissions	
<del>ctf/LFireCfg/Banned.cfg</del>	File for IP banning	
<del>ctf/LFireCfg/ChangeMap.cfg</del>	File for validating "changemap" command	
<del>ctf/LFireCfg/LHelp.txt</del>	File for "LHelp" command	
<del>ctf/LFireCfg/MapLoop.cfg</del>	File for defining the custom map loop	
<del>ctf/LFireCfg/MOTD.cfg</del>	File for defining the message of the day	
<del>ctf/LFireCfg/ScorePreload.cfg</del>	File for preloading level scores	
<del>ctf/LFireCfg/Shutup.cfg</del>	File for silencing users	
<del>ctf/LFireCfg/TextFilter.cfg</del>	<u>LFireCfg/TextFilter.cfg</u>	File for filtering words and phrases

## 2.2. Quick Start

To get started as quickly as possible, simply replace the gamex86.dll (Win32) or gamei386.so (Linux) gamex86.dll, gamei386.so, or gamesparc.so in your ctf directory and then recreate the file and directory structure that's contained in the L-Fire CTF distribution:configuration file archive.

Other than the gamex86.dll or gamei386.so,game binary, all L-Fire CTF configuration files are located in the L-Fire CTF configuration directory or a subdirectory of the configuration directory. The name of the configuration directory defaults to LFireCfg and this directory must be a subdirectory of your ctf directory. For example, on Win32 servers the full pathname to the LFireCfg directory might be c:\Quake2\ctf\LFireCfg. On Linux servers the full pathname might be /usr/games/quake2/ctf/lfirecfg.

## 2.3. Default L-Fire CTF Configuration

As explained before, the default setup of L-Fire CTF does not activate any "impure" game features like quick weapon switching or BFG banning. The new features that are active by default and that will be immediately noticeable to the players are the level start countdown, anti-spam, team

balancing, ~~VWep~~, reconnect resume option, green player names in console messages, and flag carrier names on the HUD. Any of these features can be turned off through server cvars. See the Appendix for a list of the cvars. In addition to the obvious features, idle player detection is active and is set for a default of 15 minutes.

## 2.4. Your Server Configuration

Your server's startup configuration file (usually `server.cfg`) does not need any modification to run L-Fire CTF with its default features.

# 3. Running L-Fire CTF

## 3.1. Message of the Day

The message of the day file is located in the L-Fire CTF configuration directory and is called "MOTD.cfg". The message of the day appears in an inventory window when a client first connects to the game. You can customize your message of the day so that each line appears centered, left aligned, or right aligned. Each line can also have white text or green text.

To make your own message of the day, see the instructions inside the included MOTD.cfg file.

**Note:**

The message of the day file is only read when the server is first started. If you change the message of the day while the server is running, you must issue an "sv loadmotd" command from the server console to re-read the message of the day.

## 3.2. Highscores

The server will maintain a top 15 highscore list for each level. When a player enters a level, the top 10 scores for the level are displayed on the player's HUD and the scores are removed after about 10 seconds. If the player happens to have scores on the top 10 list (determined by matching the name), the player's scores are highlighted in green. The full highscore list of 15 players is also displayed to all of the players and spectators at intermission. After the current level's scores are displayed at intermission, the highscore list then appears.

### 3.2.1. Saving Highscores

The highscores for each level are saved in the "HighScores" subdirectory of the L-Fire CTF configuration directory. Each score file has the name `<level>.txt`, where `<level>` is the level name like `q2ctf1`, `q2ctf2`, etc. For example, the highscore file for `q2ctf1` is "q2ctf1.txt".

The L-Fire CTF distribution comes with highscore files for the Id CTF levels (`q2ctf1` – `q2ctf5`) and the AllStar2 CTF levels (`as2m1` – `as2m7`). For new levels, you can either let the server run the level once, in which case the server will automatically create a new highscore file when the level ends, or you can make a highscore file by copying the "NewScores.txt" file which is included with the L-Fire CTF distribution. For example, if you have a map named "mymap", then you can make a copy of "NewScores.txt" and call it "mymap.txt". If you run a new map and do not create a highscore file for the map, then the displayed highscores will simply be blank until the list is populated with

highscores. The NewScores.txt file simply contains 15 scores by “Player” and with a score of 10 points each.

**Note:**

If you don’t want your highscore lists to start out with 15 scores of 10 points by “Player”, then simply delete all of the files in the Highscores directory. The highscore lists will then start out as 15 scores with a blank name and a score of 0 points.

**3.2.2. Enabling Highscores**

By default, the highscore feature is disabled. To enable highscores, set the “savehighscores” cvar to “1”. You should set the “fraglimit” cvar to “0” and make sure you also set the “timelimit” cvar to something like 20 or 30 minutes. Finally, you should also have a capturelimit of something like 8 captures.

**3.3. Level Start Countdown**

In a game like CTF where there are several desirables in the level such as techs, it doesn’t make much sense that people with fast computers and fast connections can start the game before other players after the server changes levels. The way to fix this problem is to have a short countdown before the game begins. Having a countdown at the start of the level gives everyone a fair chance to get into the game after the server changes levels.

During the level start countdown, all of the players will float in the level as spectators. The players’ HUDs will display a message that says “Get ready to rumble...” and the HUDs will also display the seconds remaining in the countdown. During this time, players can join teams and the scoreboard will indicate they are on a team, but they won’t actually enter the level until the countdown ends.

The countdown period is specified by the “levelstart” cvar. The default is 30 seconds. Note that almost no one gets into the game until 5-10 seconds have passed, so if you want your server to have a fair level start, this value should always be at least 15 seconds. The countdown is also a good time for players to even the teams if the previous level’s teams were unbalanced skill-wise.

**3.4. Filtering Names and Chat Messages**

L-Fire CTF will let the server admin define a list of inappropriate words and phrases (referred to from here on as text strings) that can be filtered from player names and chat messages. Text strings that are matched by the filter are replaced with a single asterisk (\*).

The text filter will allow you to define text strings that should be removed, as well as text strings that should not be removed. This is useful when an inappropriate text string may be a substring of an appropriate string. For example, the text filter will let you define the word “lag” as inappropriate, but it can be configured to ignore the word “flag”.

The most common method for modifying a text string so the filter won’t recognize it is to insert characters such as spaces or periods between the letters in the string. For example, if the word “lag” is inappropriate, then players will typically try to insert spaces between the characters so “lag” becomes “l a g”. The text filter in L-Fire CTF is sophisticated enough to automatically recognize this and replace the entire padded string with an asterisk, so you do not have to try to define every combination of a string that has been padded by inserting characters such as spaces or periods.

### 3.4.1. Defining the Text Filter

The file “TextFilter.cfg” in your L-Fire CTF configuration directory defines the words and phrases that the text filter will remove from player names and chat messages. Instructions for editing the file are inside the file itself. The file is only loaded when the server first starts. If you edit the file while the server is running, then you must issue an “sv loadtextfilter” command at the console to reload the text filter file.

### 3.4.2. Enabling Filtering

By default, name filtering and chat filtering are disabled. To enable name filtering, set the “filternames” cvar to “1”. To enable chat filtering, set the “filterchat” cvar to “1”.

## 3.5. Weaponflags

Quick weapon switching, weapons banning, and BFG suicide are controlled with the “weaponflags” cvar. See the Appendix for a description of this cvar and how to specify the flags. By default, quick weapon switching is off, no weapons are banned, and BFG suicide is off.

## 3.6. Configuring the HUD

The server admin can select one of two player HUDs. Note that the players cannot individually control their HUD.

### 3.6.1. Standard HUD

The standard HUD is exactly the same as the standard Id CTF HUD with one enhancement – the names of the flag carriers are displayed next to the flag icons in the lower right of the HUD. This lets players know with a quick glance who has the red and blue flag. If you don’t want flag carrier names on the HUD, then set the “carriernames” cvar to “0”.

### 3.6.2. Time Remaining HUD

A second HUD is available that displays the time remaining in the level in the upper right hand corner of the HUD. In order to support the time remaining display, ~~two things are removed from the standard HUD – the pickup string/icon and the blinking flag icon. has been removed from the standard HUD.~~ The pickup string and icon are displayed in the lower left of the HUD when a player picks something up. This seems pretty useless as most people don’t need to see item names and their icons unless it’s the very first time they’re playing Quake. ~~The blinking flag icon appears in the upper right when the player has the enemy flag. This is useful in standard CTF, but since L-Fire CTF displays the player’s name next to the enemy flag icon in the Quake. lower right, it’s somewhat redundant. The flag carrier will see his name on the HUD and know he has the flag. To make it even easier to notice, the player’s name will blink instead of being constant if you use the time remaining HUD.~~

The pickup string/icon ~~and the blinking flag have~~ has to be removed to support the time display on the HUD, otherwise there’s a risk of overflows as too much data is sent to the client when the level is loaded.

~~As with the standard HUD, flag carrier names can be removed from the time remaining HUD by setting the “carriernames” cvar to “0”, although this is not recommended because then there will be no indication that a player has the flag, other than checking the score list.~~

By default, the standard HUD is active. If you want to use the time remaining HUD, then set the “hudtime” cvar to “1”. Changes to the hudtime cvar only take effect after the level changes. This is to prevent possible overflows that can happen if the HUD is changed on the fly.

## 3.7. IP Banning

Clients can be banned from the server by using the IP banning feature. The Banned.cfg file in the L-Fire CTF configuration directory specifies the clients that you wish to ban. IP banning works with IP addresses and subnet masks. It does not use wildcards. Using an address and subnet mask is more flexible than using wildcards, although it is harder to understand if you don’t know how subnet masks work.

Within the Banned.cfg file, each IP address you wish to ban should be specified on a separate line that starts with an equal sign (=). Any line that does not begin with an equal sign is ignored. After the equal sign, you should specify the IP address and then the subnet mask. Include a space or TAB between the IP address and subnet mask.

Here are a couple of examples on how to use IP banning:

### 3.7.1. Ban a single host (one IP address)

Let’s say you want to ban the single host 192.168.10.40. For a single host, the subnet mask should be 255.255.255.255. The line in your Banned.cfg file would be:

```
=192.168.10.40    255.255.255.255
```

### 3.7.2. Ban a class C subnet (for example – everyone from a small company)

In this example, we want to ban all hosts with an IP address of 192.168.10.X. In this case, the subnet mask will not be 255.255.255.255, as in the previous example. For a class C subnet mask, you will use 255.255.255.0. This will ban all hosts that have an IP address beginning with 192.168.10.

```
=192.168.10.0    255.255.255.0
```

For further information on IP addresses and subnet masks, search the Internet or get a book on TCP/IP configuration.

#### Note:

The IP banning file is only read when the server is first started. If you make changes to this file while the server is running, you must issue an “sv loadban” command from the server console to re-read the list of banned clients.

## 3.8. Silencing Players

Have you ever wished you could just prevent a player from chatting – maybe because they’re just stupid or whining too much? Well now you can with the “shutup” and “unshutup” commands. When a player has been silenced, any of the player’s say or say\_team chats are echoed to his own screen as normal, but no other players will get the chat message. The server console and chat log will still show the player’s chat message, preceded by “(Shutup)”. In order to use the “shutup” and “unshutup” commands, you must have admin permission. You can use the commands from the server console by prefixing them with an “sv” – “sv shutup” and “sv unshutup”.

If you want to automatically silence a player whenever he connects to the server, you can specify the player’s name in the “Shutup.cfg” file located in the L-Fire CTF configuration directory.

Instructions for adding players to the file are contained within the file itself.

**Note:**

The player silencing file is only read when the server is first started. If you make changes to this file while the server is running, you must issue an “sv loadshutup” command from the server console to re-read the list of clients you wish to silence.

## 3.9. Logging

L-Fire CTF generates four kinds of logs: IP/Name, chat message, admin usage, and frag stats.

### 3.9.1. IP/Name

For every client that connects to your server, the time, IP address, and player name is logged to this file. If the client changes his name while he’s connected to the server, the name change is also logged. L-Fire CTF also logs refused connection attempts, such as attempts by banned players to connect to the server. By default, IP/Name logging is off. Set the “lognames” cvar to “1” to turn on IP/Name logging.

IP/Name logs are named connect\_yymmdd.log, where yymmdd is year, month, and day. The value for “dd” will only be “01” or “16”, so all connections from the 1<sup>st</sup> to the 15<sup>th</sup> of the month appear in an “01” file, and all connections from the 16<sup>th</sup> to the 31<sup>st</sup> of the month appear in a “16” file. This keeps things a little organized and prevents the log files from growing too big.

Bot detections are also written to the IP/Name log. By default, bot detection logging is turned on, but the detections will not be written unless IP/Name logging is also turned on. See the “Bot Prevention” section for more information.

### 3.9.2. Chat Message

All team and non-team chat messages can be logged. The log file will contain the time, IP address, name, team, and chat message text. By default, chat message logging is off. Set the “logchat” cvar to “1” to turn on chat message logging.

Chat message logs are named chat\_yymmdd.log, where yymmdd is year, month, and day. As with IP/Name logs, the value for “dd” will only be “01” or “16”.

### 3.9.3. Admin Usage

Any uses of admin commands are logged to admin log files. Example uses of admin commands include successful and failed attempts to gain admin permission, as well as successful and failed attempts to execute admin-only commands. For each use of an admin command, the time, IP address, name, and the command is logged. By default, admin logging is on and can be turned off by setting the “logadmin” cvar to “0”.

Admin usage logs are named admin\_yymmdd.log, where yymmdd is year, month, and day. As with IP/Name logs, the value for “dd” will only be “01” or “16”.

### 3.9.4. Frag Stats

The frag stats log is output in the Log Standard format developed by Mark Davies. This log can be read by such frag stats parsers as GibStats. By default, frag stats logging is off. Set the “logfrags” cvar to “1” to turn on frag stats logging.

There are two options for how the server saves frag stats to the log. The server can either

automatically start a new log each day at midnight, or the server can keep appending stats to the same file (not recommended).

By default, the server will start a new log file at the first level change after midnight of each day. In this case, the base name of the file is specified by the “stdlogname” cvar. The server takes the filename from stdlogname and appends the date to the filename. So for example, if the stdlogname cvar is set to “StdLog.log”, the server will create a file named “StdLog\_yymmdd.log”. Unlike the case with IP/Name, chat, and admin logs, the value for “dd” will not be limited to “01” and “16”. Instead, the value for “dd” will match the day the log file was created.

The “dailyfraglog” cvar determines whether a new log file is started each day or the server continues to append to the same file. By default, daily logging is on. Set the dailyfraglog cvar to “0” if you do not want the server to start a new frag log each day. In this case, the “stdlogname” cvar will specify the exact name of the log file, and the current date (yymmdd) will not be appended to the log filename.

For more information about the Log Standard format, see <http://www.planetquake.com/gslogmod/logstandard.html>

For more information about GibStats, see <http://www.planetquake.com/gibstats/>

### 3.9.5. Flushing Log Files

For performance reasons, IP/Name, chat message, and admin logs are only written to disk during level changes. The frag stats log is written on-the-fly and uses standard operating system file I/O buffering. You can issue an “sv flushlogs” command from the server console to flush all of the logs at any time.

## 3.10. Using Custom Map Loops

You can have the server run a custom map loop by entering the map names in the “MapLoop.cfg” file. This file is read every time a level ends to determine the next map to run.

When the server starts, the map specified at the end of your server’s configuration file (usually server.cfg) is always run first, so you might as well start your loop with the same map.

If you manually run a map that does not exist in the loop, the loop will start at the beginning when the map ends. If you manually run a map that DOES exist in the loop, then when the map ends, the server continues with the next map in the loop. For example: you have a map loop with maps named x1, x2, x3, x4, and x5. If the current map is x2, and you change the map to x4, when x4 is done, the server will continue with x5.

#### Note:

The server will only read the MapLoop.cfg file if you have set the “maploop” cvar to “1”. By default, the maploop cvar is set to 0.

## 3.11. Using Entity Files

Entity files specify the items in a map and are usually included as text files inside the actual map files

(.bsp files). L-Fire CTF supports custom entity files that can be used instead of the entity files that are built into the maps. This can be especially useful for adding CTF items such as flags and banners to non-CTF maps, or for modifying or redoing item placement in a map.

The editing and format of entity files is beyond the scope of this document. Go to some of the level editing sites on the Internet if you want to know how to edit entity files. You can also obtain entity files that have been created for playing CTF on various non-CTF maps, such as the Id single player maps and DM maps.

### 3.11.1. Loading Entity Files

Entity files are located in the “Ent” subdirectory of your L-Fire CTF configuration directory. When the server loads a map, it checks the Ent subdirectory. If it finds an entity file named <level>.ent, then the server will use the entities in the custom entity file instead of using the entity file that’s built into the map. For example, if you want to create a new entity file for q2ctf1, then your entity file should be called “q2ctf1.ent” and be placed in the Ent subdirectory.

### 3.11.2. Outlands Entity File

Included in the L-Fire CTF distribution is an entity file for q2ctf4 (Outlands). As everyone knows, red base is at a disadvantage compared to blue base if the BFG has been banned from the server. This new entity file has been designed and playtested to give the red team more of a chance in the Outlands. The new entity file adds a pack of rockets, two packs of cells, a pack of slugs, and a railgun to red base. In addition, the rocket launcher has been moved away from the respawn pad and to the bridge in order to prevent a spawn protected player from respawning, grabbing the rocket launcher, and killing red base defenders while he is spawn protected.

If you want to use the custom Outlands entity file, then you must rename the entity file from “nobfg\_q2ctf4.ent” to “q2ctf4.ent”. The file is named “nobfg\_q2ctf4.ent” in the L-Fire CTF distribution so it won’t accidentally be loaded if you don’t want to use the custom entity placement.

## 3.12. Creating Admin Users

The admin permissions feature allows you to give selected clients permission to use admin commands. A list of the admin commands can be found in the Appendix.

Each admin client can be given a unique password, and permissions for each client are individually specified. For example, you may want to give a user permission to run matches, but no other permissions. For another user, you may want to allow use of the shutup and forceteam commands, but no other. Since each client is given an independent list of permissions, and the permissions themselves are very specific, you have a lot of flexibility in controlling how admin commands are used on your server.

### 3.12.1. Permissions

The permissions for each client are specified by creating a string of permission flags. The flags are given below:

\* Wildcard that assigns all admin permissions

- A Change dmflags cvar (setdmflags command)
- B Change weaponflags cvar (setweaponflags command)
- C Change fraglimit cvar (setfraglimit command)
- D Change capturelimit cvars (setcapturelimit/setmercylimit commands)

E	Change timelimit cvar	(settimelimit command)
F	Change antispam cvar	(setantispam command)
G	Change teambalance cvar	(setteambalance command)
H	Change spawnprotect cvar	(setspawnprotect command)
I	Change minping cvar	(setminping command)
J	Change maxping cvar	(setmaxping command)
K	Change password cvar	(setpassword/unsetpassword commands)
L	Change overtime cvars	(setot_style/setot_rarounds/setot_sdtimelimit commands)
M	Change colorquad cvar	(setcolorquad command)
N	Change hudtime cvar	(sethudtime command)
<u>O</u>	<u>Change offhandhook cvar</u>	<u>(setoffhandhook command)</u>
P	Allow use of match commands	
Q	Allow use of changemap command	
R	Allow use of shutup command	
S	Allow use of forceteam command	
T	Allow use of boot command	
U	Allow use of bootban command	
V	Allow use of bootlevelban command	
W	Allow use of bootfullban command	
X	Ignore ping restrictions	
<u>Y</u>	<u>Change itemflags cvar</u>	<u>(setitemflags command)</u>

### 3.12.2. Adding Admin Users

In order to add admin users, you add passwords and permissions to the Admin.cfg file in your L-Fire CTF configuration directory. Each admin line must start with an equal sign (=). The first string up to a slash (/) is considered the username and password. You don't have to have a separate username and password. You can just use a single word, or a whole sentence, or whatever you like. Note that the username/password is case-insensitive.

The string of letters after the slash defines the permissions. Instead of using a set of binary flags like the way dmflags and weaponflags work, the admin permissions are defined by a string of letters. Each letter corresponds to a specific permission. To build a set of permissions, just add the letter to the permissions string. For example, to give someone permission to run matches and allow them to use the forceteam command, the permissions string would be "PS" - the letter "P" gives the user admin permission to run matches, and the letter "S" allows the use of the forceteam command. It's that easy.

You can duplicate letters in the permission string and the letters can appear in any order. Any characters that are not an asterisk or the letters A through ~~X~~Y are simply ignored.

### 3.12.3. Admin Command

The client uses the "admin" command to gain administration access. For example, if the admin password is "mypassword", the client would type "admin mypassword" at the console to gain his admin permission.

For convenience, the client can set his "admin\_password" cvar to his password, then he only has to type "admin" to gain his admin permissions. For example, if the client's admin password is "mypassword", he could put the line "set admin\_password mypassword" in his autoexec.cfg file and then simply type "admin" when he connects to the server.

Once a client has gained admin access, typing “admin” without a password will display a summary of the admin permissions that have been granted to the client.

Here are some examples for admin users:

#### 3.12.4. Superuser (all permissions)

This admin has two separate words for the password: superuser and mypassword. The user types “admin superuser mypassword” at the console to get the given permissions. In this case, all of the permissions will be assigned because the asterisk is used.

```
=superuser mypassword/ *
```

#### 3.12.5. Match Administrator

A match administrator needs permission to use the match commands. In this example, the username/password is just a single word “password321”. The match administrator would type “admin password321” at the console to gain permission to use match commands. The permission flag “P” is for match administration.

```
=password321/ P
```

#### 3.12.6. Experienced HPB

In this example, let’s assume you’re running an LPB-only server with ping restriction, but you want to allow an experienced HPB to play. In this case, you can give the player the admin permission to avoid ping restrictions. The permission flag “X” allows a player to ignore ping restrictions.

```
=goodhpb/ X
```

### 3.13. Restricting Player Pings

L-Fire CTF includes ping restriction to enable you to enforce minimum and/or maximum ping requirements to play on the server. Use the “minping” cvar to specify a minimum ping, and use the “maxping” cvar to specify a maximum ping. By default, both minping and maxping are set to 0, which disables minimum and maximum ping checking.

The server bases its ping checking on a player’s average ping. Pings are not checked until the player has been connected to the server for at least one minute. If a player does not meet the ping requirements, then he is automatically moved to spectator and prevented from joining a team.

If you want to make an exception for a player who does not normally meet the ping requirements, then you can give the player an admin permission of “ignore ping restrictions”. Such might be the case, for example, to allow an experienced HPB player to play on an LPB-restricted server.

### 3.14. Bot Prevention

L-Fire CTF includes several bot prevention and detection features. By default, client connections on port 27902 (default ZBot port) are refused. This prevents newbie ZBot players from connecting to the server. Other ZBots are detected by two separate detection methods. The two methods are used to prevent problems with false positives that plague first-generation ZBot detection mods. Once a bot has been detected, the bot can be logged, a warning message can be displayed to the players’ consoles, the bot can be booted from the server, and the player’s IP can be automatically banned.

Port banning is controlled by the “bannedport” cvar, which defaults to “27902”. Bot detection is controlled by the “logbots” and “detectbots” cvars. By default, “logbots” is ON so ZBot detections

will be logged to the IP/Name log. Note that the “lognames” cvar must also be ON, otherwise ZBot detections will not be logged. The “detectbots” cvar controls bot detection. A value of “1” (the default) will detect bots and display a message to everyone’s console. A value of “2” will detect bots and also boot them from the server. A value of “3” will detect bots, boot them, and also ban the player’s IP address until the server is restarted. The server admin will have to add the IP address to the banned.cfg file for a permanent ban.

### 3.15. Server Maximum Rate

On the client side, the “rate” cvar is used to specify the maximum number of bytes per second that the server will send to the client. There is no built-in Id command to control maximum rate from the server’s side. Since the default client rate is 25000, a group of LPBs on a high bandwidth connection such as a T3 can quickly use all of your server’s bandwidth, especially if your server is on a T1 line or less.

L-Fire CTF has a “maxrate” cvar that limits a client’s maximum rate at the server, regardless of a client’s higher rate setting. For example, if the client’s rate is 25000 and the server maxrate is 15000, a maximum of 15000 bytes per second will be sent to the client. If the client’s rate is lower than the server’s maxrate of 15000, then the lower client rate is used, as expected. Unlike other mods that do server side rate limiting, L-Fire CTF will NOT force the server’s maxrate to the client side. Instead, the rate is limited at the server side, which is much better because it doesn’t change the client’s rate behind their back and without their permission.

Unless your server is on a very high bandwidth connection, a recommended value for maxrate would be 10000 – 15000. On the client side, modem users are supposed to set a maxrate of 2500, single B-channel ISDN 5000, dual B-channel ISDN and cable modems 10000, and a T1 25000. This gives you an idea that a server maxrate of 10000 – 15000 is more than reasonable and will help if your server has limited bandwidth.

### 3.16. Additional Features

This section includes some recommended L-Fire CTF features that are turned off by default because they change Id’s CTF in a minor way.

#### 3.16.1. Team Colored Quad

~~This feature changes the color of the quad glow so that a red player with quad will glow red, and a blue player with quad will glow blue. Players that are respawn protected or that have invulnerability will glow yellow. This feature makes it much easier to tell what team a player is on when the player has quad. In addition, players with quad will no longer blink, so there is less of a problem with players rounding a corner, running into someone with quad, and not realizing it because the quad glow was in an off blink cycle. The quad glow will blink quickly only when the quad is almost out. To activate team colored quad, set the “colorquad” cvar to “1”.~~

#### 3.16.2. Obituary Filtering

Obituary filtering limits an individual player’s obituary messages to who that player killed, and who killed the player. Important messages such as flags being stolen and flag returns are displayed as normal. This feature is especially useful on busy servers where the constant stream of (nearly useless) obituary messages interferes with the easy reading of chat messages. To activate obituary filtering, set the “filterobits” cvar to “1”. Whether server obituary filtering is active or not, players

can override the settings for themselves by using the “fullobits” and “limitobits” commands. Note also that obituary filtering does not affect spectators and obituary filtering is automatically turned off during matches.

### 3.16.3. Respawn Protection

Respawn protection is commonly used on public servers to prevent lamers from camping across respawn points with (usually) a railgun. When a player respawns, they are invulnerable for a couple seconds, which gives the player enough time to get off the respawn pad and start running around. This makes it nearly impossible for a camping lamer to get easy railgun kills. If the respawned player picks up the enemy flag or picks up a weapon, he immediately loses respawn protection. This prevents abuses in some levels where powerful weapons are located immediately next to respawn points (Outlands red base rocket launcher and blue base hyperblaster in particular). The “spawnprotect” cvar specifies the number of seconds that a player is protected after he respawns. One or two seconds is a good value for respawn protection.

## 4. Overtime

One of the weaknesses in CTF is when teams are tied in captures at the end of the level and the team with more total points wins. Winning a tied CTF game based on total points is like winning a tied football game based on total yards. Stats like frags and total yards have nothing to do with determining who should win when it’s a team game.

L-Fire CTF implements two versions of overtime: rocket arena overtime and sudden death overtime, both of which are explained below.

### 4.1. Rocket Arena Overtime

This is an overtime based on pickup games that you see in the Quake Rocket Arena mod. After regulation ends, players are moved to spectator and all items, flags, and techs are removed from the level. After a short countdown, each team’s players are spawned into the level at random locations throughout the level. Players start with 150 health, 150 armor, all weapons (except the BFG), and plenty of ammo. Any player who is killed is removed from the level and must wait as a spectator until the current round ends. The last team left standing wins the round and is awarded a capture.

Rocket arena overtime implements a couple of features that are different from the CTF game and different from the Rocket Arena mod. First, quick weapon switching is automatically enabled, regardless of the server’s normal quick weapon switching setting. Second, there is a player health countdown. Players lose one unit of health per second. If a player’s health drops to zero, he dies and the message “... ran out of health” is displayed. If a player kills someone else, ~~all of his health, armor, and ammo are fully restored. he is given a kill bonus of +50 health and +50 armor.~~ The health drain discourages camping and encourages players to engage the enemy while their health is close to full strength. The longer players wait to engage the enemy at the start of the overtime, the less health they will have.

Rocket arena overtime is designed for public servers and is probably not appropriate for matches. To use rocket arena overtime, set the “ot\_style” cvar to “1”. You can control the number of rocket arena rounds using the “ot\_rarounds” cvar, which defaults to “3”.

## 4.2. Sudden Death Overtime

This is a “first cap wins” overtime and is designed for both public servers and matches. After regulation ends, players are moved to spectator and the level is reset. After a short countdown, the players are spawned in their respective bases. The first team to capture the enemy flag wins the level.

During sudden death overtime, there are no techs, no quad, and no bonus points are awarded for such things as base defense, flag carrier frags, flag returns, etc. Eliminating techs gets rid of the luck factor at the beginning of the level, eliminating quad prevents the overtime from turning into a “first team who gets quad wins” game, and eliminating bonus points prevents teams from simply camping in their base to try and rack up defensive bonus points. So sudden death overtime is designed as more of a skill game than regulation.

If the sudden death overtime timelimit is reached, then the team with more frags will win the overtime. Only overtime frags are counted – frags gained during regulation are ignored. Unfortunately, we have to use frags to determine the winner, even though frags shouldn’t matter in a team game. If we don’t use frags and a timelimit, then the overtime might go on forever.

To use sudden death overtime, set the “ot\_style” cvar to “2”. The sudden death timelimit is specified using the “ot\_sdtimelimit” cvar. By default, the timelimit is 10 minutes.

## 5. Running Matches

L-Fire CTF has features that make it much easier to run organized matches than if you were using the standard Id CTF. In order to activate the match options, you must set the “matchoptions” cvar to “1”. By default, match features are not enabled.

Five primary commands are used to manage a match: matchsetup, matchstart, matchend, matchstate, and changemap. In addition, the lockteams, unlockteams, lockchat, and unlockchat commands are available to help control matches that may take place on public servers.

In order to use match commands, a user must have the necessary admin permission. See the above section on creating admin users for information on giving users permission to use match commands. Alternatively, elections can be used to elect a match admin (described in the next section).

### 5.1. Match Admin Elections

Elections can be held to elect a client as match admin. To start an election, use the “elect” command. Using “elect” with no arguments will start an election for the player who issued the command. Specifying a player name or player number will start an election for the specified player.

Once an election has started, all clients are requested to vote. Clients type “vote yes” or “vote no” to cast their ballot in the election. The voting period last for a default of 30 seconds, after which the votes are counted. If there were enough “yes” votes, then the candidate is given a set of admin permissions that are appropriate for running a match. Note that any client that does not vote will automatically have a “no” vote cast.

The admin permissions that are assigned to an elected client are as follows: permission to change

dmflags, weaponflags, fraglimit, timelimit, capturelimit, spawnprotect, minping, maxping, ot\_style, ot\_rarounds, ot\_sdtimelimit, hudtime, colorquad, and use the full set of match commands as well as the forceteam command.

By default, match elections are disabled. Enable match elections by setting the “matchvote\_allow” cvar to “1”. The minimum number of clients required to hold an election, the percentage of “yes” votes required to elect the candidate, the voting period, and the maximum self-election attempts per level can be controlled using the “matchvote\_x” cvars. See the Appendix for more details.

## 5.2. Setup Stage

To start a match, use the “matchsetup” command. Once this command is given, the server switches to match mode and the server runs in the match setup stage. In this stage, all items on the current level are respawned, dropped weapons, ammo, and other items are cleared, bodies and gibs are removed, flags are returned, and flag captures are set to zero. This lets you start the match on a clean level without having to do a full level restart. When the match starts and players actually enter the level, the techs are respawned to new positions so no player is given an advantage by knowing where the techs are located.

In the setup stage, the server also removes all players from the level so they float as spectators. During this time, players can join teams and the scoreboard will indicate they are on a team, but they won't actually enter the level until the match begins.

## 5.3. Countdown

Once everyone has selected a team and is ready to begin the match, the “matchstart” command can be given. At this time, a countdown timer is displayed on the player HUDs, and when the countdown reaches zero, players are spawned into the level on their chosen teams and the fighting begins.

## 5.4. Ending the Match

To end the match, use the “matchend” command. If the server is currently in the match setup or countdown stage and there are no match administrators connected (if, for example, the match administrator disconnected), then the server automatically switches to non-match mode. This is done to prevent players from being stuck in the setup stage. If the match is currently in progress (game on), then the server will stay in match mode until the end of the level – even if there are no match administrators connected. This is done so server overrides stay in effect during the match (see the section below on server overrides).

Note that when the level ends (due to timelimit, capturelimit, etc.), the scores are displayed but the server will not automatically change to the next level. This allows players to get screenshots of their scores, chat, etc. The level can be changed using the normal methods such as the changemap command or using rcon.

## 5.5. Server Feature Overrides

In order to prevent non-match features from interfering with a match, the server automatically

disables anti-spam, team balancing, ping checking, name filtering, chat filtering, and obituary filtering, ~~and the “mercylimit” cvar (the mercy rule)~~. In addition, the reconnect resume option is automatically enabled. When the match ends, all of the overridden features are restored to their non-match values.

## 5.6. Locking the Match

Once players have selected their teams and you're ready to begin the match, you may want to lock down the teams or even the entire server to prevent interruptions, such as spectators joining a team or other people connecting to the server.

### 5.6.1. Locking Teams

You can prevent players from changing teams or prevent spectators from joining a team by using the “lockteams” command. Teams will be locked until the “unlockteams” command is issued. Teams are also automatically unlocked when the server exits match mode.

### 5.6.2. Locking Spectator Chat

To prevent spectators from chatting to the players, use the “lockchat” command. Spectators will be prevented from chatting to the players, but they'll still be able to use say\_team or messagemode2 to chat to other spectators. Chat locking only takes effect when the match is actually in progress. During match setup and countdown stages, spectators can chat to the players as usual. As with team locking, spectator chatting is automatically unlocked when the server exits match mode. Use “unlockchat” if you want to manually unlock spectator chatting.

### 5.6.3. Locking the Server

You can prevent players from connecting to the server by using the “setpassword” and “unsetpassword” commands. Using setpassword works the same way as setting the server's password cvar, but you can also use unsetpassword to conveniently clear the password. The unsetpassword command is necessary because Quake does not provide an easy way of remotely clearing a cvar once it's been set to some value. Note that using setpassword/unsetpassword requires the appropriate admin permission. Unlike the case with team locking and chat locking, the server password is NOT automatically cleared when the match ends. The setpassword and unsetpassword commands actually work to simply set the server's password cvar, and they function independently of the server's match mode. In other words, you can use setpassword/unsetpassword whether you're in match mode or not.

## 6. Appendix

### 6.1. Specifying Player Names

Several of the L-Fire CTF commands (ignore, forceteam, shutup, and boot, for example) take either a player name or a player number. The player number can be obtained by using the “players” command. When using the player's name, the server will recognize partial name matches. For example, if you have a player named “JoeLamer” that you wish to ignore, you can type “ignore joelamer”, “ignore joe”, “ignore lamer”, etc. If more than one player matches the name, then the server will list all of the players that match. For example, if there is also a player named “JoeCamel” and you type “ignore joe”, the server will list the two players that match “joe” along with the player numbers. You can then try the command again using a more specific name, or you can just use the

player number.

## 6.2. Client Commands

These commands can be issued by a client from the client's console. There are no restrictions on who can use these commands.

### 6.2.1. censor

Turns on chat censoring for the client, regardless of the server's filterchat setting. Note that this command will only turn on censoring for the specific client. No other clients will be affected.

### 6.2.2. chase [player\_name / player\_number]

Using the chase command with no arguments moves a player to spectator (if he's not already a spectator) and goes directly into chase camera mode. If a player's name or number is specified, then that player will be chased. This makes it a lot easier to chase a certain player without having to cycle through the players one at a time.

### 6.2.3. coords

Displays the X, Y, and Z coordinates for the player's current position in the map. This is mostly useful for a server operator who is designing custom entity files.

### 6.2.4. elect [player\_name / player\_number]

Starts a match admin election for the specified player. If no player is given, then an election is started for the player using the command.

### 6.2.5. fragstats

Executes the "fullobit" and "lowtext" commands so the console output for the player will be compatible with frag stats parsers such as GibStats. This command does nothing more than execute the two commands.

### 6.2.6. fullobit (fullobits)

Overrides the "filterobits" cvar for the player so the player will always receive full obituary messages. This is useful if the player is saving the console output for later parsing by a stats parser such as GibStats.

### 6.2.7. highscores [level]

Displays the highscores for the specified level. The highscores feature must be active.

### 6.2.8. hightext

Allows console messages for a player to be highlighted with green text. This command undoes the "lowtext" command. Note that even by using this command, a player cannot get highlighted console messages if the server has "greennames" set to "0".

### 6.2.9. ignore [player\_name / player\_number]

Allows a player to ignore another player. When a player is ignored, the ignoring player still sees the ignored player's chat messages, but they appear in plain white text and the message is not accompanied by a triple beep.

### 6.2.10. lhelp (LHelp, info)

Displays the text contained in the LHelp.txt file in the server's L-Fire CTF configuration directory.

**6.2.11. limitobit (limitobits)**

Overrides the “filterobits” cvar for the player so the player will only get limited obituary messages – who the player killed and who killed the player. Obituary limiting only affects players – spectators always receive full obituary messages.

**6.2.12. lowtext**

Overrides the “greennames” cvar for the player so console text will not have green names or messages. This is useful when the client is saving the console log for later use with a frag parser like GibStats. If green text is not disabled, then the console log will contain gibberish characters that represent the green text.

**6.2.13. nocensor**

Turns off chat censoring for the client, regardless of the server’s filterchat setting. Note that this command will only turn off censoring for the specific client. No other clients will be affected. Using “nocensor” does not mean the client’s own chat messages will not be filtered, it means the client will receive uncensored chat messages from other players. His own chat messages will still be censored if the server is normally filtering chat messages.

**6.2.14. observe (observer)**

Changes the player to a spectator/observer.

**6.2.15. players**

Displays a list of the players and spectators, along with the player number. The player number can be used with various L-Fire CTF commands (ignore, unignore, and shutup for example).

**6.2.16. scores [level]**

Displays the scores for the last time the specified level was played. The “savescores” feature must be active. The server only saves scores for the most recent time a particular level was played. When a level is played again, the new scores overwrite the older scores.

**6.2.17. settings**

Displays the current server settings, such as the fraglimit, timelimit, etc. [Settings that change the default id gameplay are highlighted in green text.](#)

**6.2.18. team [red, blue, spec]**

This is an enhancement to the “team” command. A player can now specify “spec”, “spectator”, “observer”, or “observe” to switch to a spectator.

**6.2.19. time**

Displays the time remaining in the current level. Displays “no timelimit” if the server does not have a timelimit set.

**6.2.20. unignore [player\_name / player\_number]**

Allows a player to unignore another player. Chat messages from the unignored player will then be displayed as usual.

**6.2.21. vote [yes / no]**

Records a “yes” or “no” vote for the current match admin election.

## 6.3. Match Commands

These commands are used for administering matches. Note that a client must have match administrator permission in order to use any of these commands. In addition, the “matchoptions” cvar must be set to 1 in order to use any match commands.

### 6.3.1. changemap [map]

Changes the current map. All map changing is validated by the ChangeMap.cfg file to prevent attempts to change to a non-existent map.

### 6.3.2. lockchat

Prevents spectators from chatting to the players. Spectators can still chat to other spectators by using team talk (say\_team or messagemode2). Note that spectators can still chat to players during the match setup and countdown stages. Chat locking is only in effect while the match is actually being played.

### 6.3.3. lockteams

Locks the teams so players cannot change teams and spectators cannot join a team.

### 6.3.4. matchend

Takes the server out of match mode. Note that this command has no effect on the matchoptions cvar. Using matchend simply puts the server back into non-match mode. A matchsetup command can be issued later to prepare for another match.

### 6.3.5. matchsetup

This command is used to put the server into match setup mode. The server will be in the “setup” stage where players are moved to spectator, but the players can join teams and prepare for the match. This command can be issued at any time, even if the server is counting down or the match is in progress.

### 6.3.6. matchstart [time\_secs]

Starts the countdown to begin the match. This command can only be used if the server is in the “setup” stage. By default, the server countdown is 15 seconds.

### 6.3.7. matchstate

Displays the match mode: none, setup, entering match, or game on.

### 6.3.8. nextmap

Forces the server to execute the next map in the server’s map loop.

### 6.3.9. settings

Displays the current server settings, such as the fraglimit, timelimit, etc. This command is also listed in the “Client Commands” section, but its primary purpose is for verifying server settings during a match.

### 6.3.10. unlockchat

Allows spectators to chat again after spectator chatting has been locked by using the “lockchat” command.

### 6.3.11. unlockteams

Allows players to change teams again after teams have been locked using the “lockteams” command.

## 6.4. Admin Commands

These commands can only be used by clients with the correct admin permission.

### 6.4.1. admin [password]

This command is used by a client who wants to gain administration access.

For convenience, the client can set his “admin\_password” cvar to his password – then he only has to type “admin” to gain his admin permissions. For example, if the client’s admin password is “mypassword”, he could put the line “set admin\_password mypassword” in his autoexec.cfg file and then simply type “admin” when he connects to the server.

Once a client has gained admin access, typing “admin” without a password will display a summary of the admin permissions that have been granted to the client.

### 6.4.2. boot [player\_name / player\_number]

Exactly like the server console “kick” command. The word “kick” could not be used as the client command because Quake reserves the command for itself. You may specify the player by name or by number.

### 6.4.3. bootban [player\_name / player\_number]

Boots a player and simultaneously bans the player for the number of seconds specified by the “tempbantime” cvar.

### 6.4.4. bootfullban [player\_name / player\_number]

Boots a player and simultaneously bans the player until the server is restarted. This command does not put the player in the Banned.cfg file. You still need to edit the file manually in order to permanently ban a player.

### 6.4.5. bootlevelban [player\_name / player\_number]

Boots a player and simultaneously bans the player until the current level ends.

### 6.4.6. changemap [map]

This is the same as the “changemap” command listed under the “Match Commands” section. A match administrator is automatically given access to the changemap command, but a client who is given “changemap” permission does not necessarily have permission to run matches.

### 6.4.7. forceteam [player\_name / player\_number] [red/blue/spec]

Allows an admin user to force another player to the red team, blue team, or to spectator.

### 6.4.8. nextmap

This is the same as the “nextmap” command listed under the “Match Commands” section. A match administrator is automatically given access to the nextmap command, but a client who is given changemap/nextmap permission does not necessarily have permission to run matches.

### 6.4.9. setantispam [value]

Displays and sets the antispam cvar.

### 6.4.10. setcapturelimit [value]

Displays and sets the capturelimit cvar.

**6.4.11. setcolorquad [value]**

Displays and sets the colorquad cvar.

**6.4.12. setdmflags [flags]**

Displays and sets the dmflags cvar.

**6.4.13. setfraglimit [value]**

Displays and sets the fraglimit cvar.

**6.4.14. sethudtime [value]**

Displays and sets the hudtime cvar.

**6.4.15. setitemflags [value]**

Displays and sets the itemflags cvar.

**6.4.16. setmaxping [value]**

Displays and sets the maxping cvar.

**6.4.17. setmercylimit [value]**

Displays and sets the mercylimit cvar.

**6.4.18. setminping [value]**

Displays and sets the minping cvar.

**6.4.19. setoffhandhook [value]**

Displays and sets the offhandhook cvar.

**6.4.20. setot\_rarounds [value]**

Displays and sets the ot\_rarounds cvar.

**6.4.21. setot\_sdtimelimit [value]**

Displays and sets the ot\_sdtimelimit cvar.

**6.4.22. setot\_style [value]**

Displays and sets the ot\_style cvar.

**6.4.23. setpassword [value]**

Displays and sets the password cvar.

**6.4.24. setspawnprotect [value]**

Displays and sets the spawnprotect cvar.

**6.4.25. setteambalance [value]**

Displays and sets the teambalance cvar.

**6.4.26. settimelimit [value]**

Displays and sets the timelimit cvar.

**6.4.27. setweaponflags [flags]**

Displays and sets the weaponflags cvar.

**6.4.28. `shutup` [player\_name / player\_number]**

This command allows an admin user to silence another player by specifying the player name or player number.

**6.4.29. `unsetpassword`**

Clears the password cvar.

**6.4.30. `unshutup` [player\_name / player\_number]**

This command allows an admin user to unsilence a player that has been previously silenced with the “shutup” command.

## 6.5. Server Commands

These commands are issued at the server console (or through rcon). The commands must be prefixed with an “sv”, whether they are issued from the server console or as an rcon command.

**6.5.1. `sv flushlogs`**

Flushes all of the logs that may be in use (IP/name, chat, admin, and frag). For performance reasons, IP/name, chat, and admin logging does not do any file I/O while a game is in progress. These three logs are normally written to disk when the level ends or the server is shut down. The frag log is written as the game is in progress using standard operating system file I/O buffering. The “sv flushlogs” command will flush the operating system buffers for the frag log, as well as write the other three logs.

**6.5.2. `sv boot` [player\_name / player\_number]**

Same as the admin “boot” command.

**6.5.3. `sv bootban` [player\_name / player\_number]**

Same as the admin “bootban” command.

**6.5.4. `sv bootfullban` [player\_name / player\_number]**

Same as the admin “bootfullban” command.

**6.5.5. `sv bootlevelban` [player\_name / player\_number]**

Same as the admin “bootlevelban” command.

**6.5.6. `sv loadban`**

Reloads the IP banning file (Banned.cfg).

**6.5.7. `sv loadmotd`**

Reloads the message of the day file (MOTD.cfg).

**6.5.8. `sv loadshutup`**

Reloads the player shutup file (Shutup.cfg).

**6.5.9. `sv loadtextfilter`**

Reloads the text filter file (TextFilter.cfg).

**6.5.10. `sv shutdown`**

Shuts down the server and sends a “reconnect” to all of the players. When the server is restarted, the players will automatically connect, assuming they haven’t shut down Quake or connected to another

server. This command is useful when you want to do a quick server restart for some reason, and you want the players to be conveniently reconnected as soon as the server is restarted.

#### **6.5.11. sv shutdown [player\_name / player\_number]**

Same as the admin “shutdown” command.

#### **6.5.12. sv unshutdown [player\_name / player\_number]**

Same as the admin “unshutdown” command.

## **6.6. Server CVARS**

This is a list of the L-Fire CTF CVARS and a description of what each one does.

#### **6.6.1. allowresume [0/1]**

Controls whether players are given an option to resume when they reconnect to the server after having disconnected. Defaults to ON.

#### **6.6.2. allowsay [0/1]**

Controls whether players are allowed to use global chatting (say or messagemode). This cvar does not affect team chatting (say\_team or messagemode2). Defaults to ON.

#### **6.6.3. allowsayteam [0/1]**

Controls whether players are allowed to use team chatting (say\_team or messagemode2). This cvar does not affect global chatting (say or messagemode). Defaults to ON.

#### **6.6.4. antinameflood [0/1]**

Controls anti-name flooding. If a client attempts to change their name twice within half a second, they will be booted from the server. Changing names twice in half a second is only possible if an alias is being used or multiple “set name” commands are issued on a single line. Name change flooding is an insidious way to cause clients with slower connections to overflow. Defaults to ON.

#### **6.6.5. antiteamflood [0/1]**

Controls anti-team flooding. Prevents team change flooding by preventing players from changing teams more than once every three seconds. Defaults to ON. See also the “maxteamchange” cvar.

#### **6.6.6. antispam [0/1]**

Controls anti-spam checking. Defaults to ON.

#### **6.6.7. antispam\_gagtime [time\_sec]**

Specifies how long (in seconds) a player is prevented from chatting after he has been gagged by the spam checker. Defaults to 20 seconds.

#### **6.6.8. antispam\_message [message]**

Specifies the message that is displayed when a player is gagged by the spam checker. Defaults to “Sorry guys, I’m a lamer and I talk too much”. If you do not want a message displayed when a player is gagged, then set this cvar to an empty string: set antispam\_message “”.

#### **6.6.9. antispam\_teamduplicate [count]**

Specifies the number of duplicate messages that can be sent within three seconds using team talk. Additional duplicate messages within three seconds are ignored by the spam checker. For example, if this value is set to 1, players will be able to send a team message and then immediately repeat it

one additional time. If the player tries to send the same message a third time within three seconds, the message is simply ignored. Defaults to 1 (allowing two identical messages within three seconds).

#### **6.6.10. antispam\_warnings [count]**

Specifies the number of anti-spam warnings a player gets before he is automatically booted from the server. For example, if players are given two warnings, on the third spam violation, they will be booted from the server. Warnings are reset to zero after each level change. Defaults to 2 warnings.

#### **6.6.11. bannedport [port]**

Specifies a client port that is prevented from connecting. This is most useful for preventing newbie ZBots from connecting. Defaults to 27902 (the default ZBot port).

#### **6.6.12. carriernames [0/1]**

Controls whether the names of the flag carriers will appear next to the flag icons in the lower-right corner of the HUD. Defaults to ON.

#### **6.6.13. clanoptions [0/1]**

Replaced with the matchoptions cvar.

#### **6.6.14. colorquad [0/1]**

Controls whether the quad glow will match a player's team color. When this option is active, a red player with quad will glow red, and a blue player with quad will glow blue. Players that are respawn protected or have invulnerability will glow yellow. In addition, players with quad will have a constant glow and will not blink until quad is almost out. Defaults to ~~OFF~~ON.

#### **6.6.15. configdir [dir]**

Specifies the root directory for the configuration files. This directory must be a subdirectory of your quake2/ctf directory, for example, c:\Quake2\ctf\LFireCfg. Defaults to "LFireCfg". This cvar is particularly useful if you want to run more than one L-Fire CTF server from the same quake2 directory. Each server should have a different server.cfg which specifies a different configuration directory. For example, you might have a server1.cfg and a server2.cfg. The server1.cfg includes the following line: "set configdir LFireCfg1". The server2.cfg includes the following line: "set configdir LFireCfg2". Each configuration directory should have the same set of files and subdirectories—Admin.cfg, MOTD.cfg, ChangeMap.cfg, Ent, HighScores, Scores, Logs, etc.

#### **6.6.16. consolesay [0/1]**

Controls whether non-commands typed directly in the console are treated as chat messages. Defaults to OFF. It's recommended that you keep this cvar off if you're using admin permissions or use rcon from a client. This prevents mistyped "admin" and "rcon" commands from being echoed to the world along with the password if they are mistyped. In addition, disabling console chatting cuts down on accidental spam in the form of misconfigured aliases and key bindings.

#### **6.6.17. dailyfraglog [0/1]**

Controls whether the server will automatically start a new frag log at the first level change after midnight of each day. Defaults to ON.

#### **6.6.18. detectbots [0/1/2/3]**

Controls whether the server detects and optionally boots and bans bot players. A value of "1" will detect bots and display a message to all of the players' consoles. A value of "2" will detect bots and also boot the player from the server. A value of "3" will detect bots, boot the player, and also ban them until the server is restarted. Defaults to "1".

**6.6.19. filterchat [0/1]**

Controls whether chat messages are passed through the text filter. Defaults to OFF.

**6.6.20. filterchatoverride [0/1]**

Controls whether individual players are allowed to override chat filtering so they can see uncensored chat messages for themselves. Defaults to ON. There is no reason to turn this off, unless you absolutely insist that your users should not be allowed to see profanity – in which case you must be from some ultra-conservative strict country or something and then it's unclear why you'd be allowed to play Quake anyhow.

**6.6.21. filternames [0/1]**

Controls whether player names are passed through the text filter. Defaults to OFF.

**6.6.22. filterobits [0/1]**

Controls whether player obituary messages will be limited to whom the player killed, and who killed the player. This significantly reduces the number of console messages and makes it much easier to read chat messages. Defaults to OFF. Players can override this setting for themselves by using the "fullobit" and "limitobit" commands.

**6.6.23. gamestats**

This cvar displays stats on the current level and is automatically updated by the server as the game progresses. This allows people to see a general status of matches by querying the server with programs like GameSpy.

The cvar value appears as follows: "mm:ss rc/rs bc/bs r/b/s"

mm    minutes remaining  
ss    seconds remaining  
rc    red team flag captures  
rs    red team total score  
bc    blue team flag captures  
bs    blue team total score  
r     number of players on red team  
b     number of players on blue team  
s     number of spectators

**6.6.24. gibcount [count]**

Specifies the number of gibs that are thrown when a body is gibbed. Defaults to 4.

**6.6.25. grapplefix [0/1]**

Controls whether an improved grappling hook is used that eliminates the notorious bounceback and random throw behavior of Id's grappling hook. Other than eliminating the random throw, the improved grapple does not change the grapple in any way. Grapple speed and feel is exactly the same as before. Defaults to ON.

**6.6.26. grapplefly [speed]**

Specifies the speed of the grappling hook in flight – the speed of the hook before it hits a solid object and starts pulling the player. Defaults to 650.

**6.6.27. grapplepull [speed]**

Specifies the speed that the player is pulled by the grapple. Defaults to 650.

**6.6.28. grapplesky [0/1]**

Controls whether the grappling hook can grapple the sky. Defaults to OFF.

**6.6.29. greennames [0/1]**

Controls whether player names will be displayed in green text. For death messages, the player doing the killing has his name displayed in green. Defaults to ON. Players can override this setting for themselves by using the “lowtext” and “hightext” commands.

**6.6.30. hb\_effects [0/1]**

Controls whether the hyperblaster bolts will light up their surroundings. These lighting effects are one cause of hyperblaster lag, and by turning them off, rendering lag can be reduced. In practice, players will never notice that the lighting has been turned off for the hyperblaster bolts, so it’s highly recommended that you leave the hyperblaster lighting effects turned off. Defaults to OFF (no lighting effects).

**6.6.31. hb\_skipbolts [0/1/2]**

Controls whether the hyperblaster will skip the sending of some of the hyperblaster bolts. By skipping bolts, network and rendering lag are reduced because fewer entities are sent to the clients. Defaults to 0 (no bolts are skipped). Set this to “1” to not send every other bolt. Set this to “2” to not render 2 out of 3 bolts (even less lag, but the more sparse bolt train becomes more noticeable to the players.)

**6.6.32. hudtime [0/1]**

Controls whether the optional player HUD with time remaining is used. Defaults to OFF. Changes to this cvar only take effect when the level changes (this avoids the risk of overflows that may occur if the HUD is changed on the fly).

**6.6.33. idledetect [time\_min]**

Specifies how long (in minutes) a player can be idle before he is automatically moved to spectator. Defaults to 15 minutes. A value of 0 turns off idle detection.

**6.6.34. itemflags [flags]**

Specifies changes to item behavior.

<u>IF_QUAD_WINDOW</u>	<u>1</u>	<u>(quad respawns in a 24 second window)</u>
<u>IF_PA_WINDOW</u>	<u>2</u>	<u>(power armor respawns in a 24 second window)</u>
<u>IF_RANDOM_FIRST_QUAD</u>	<u>4</u>	<u>(first quad spawns in 1-60 seconds)</u>
<u>IF_RANDOM_FIRST_PA</u>	<u>8</u>	<u>(first power armor spawns in 1-60 seconds)</u>

The itemflags cvar works like the standard dmflags cvar: you pick the flags you want and add their values together to get the final itemflags value. For example, if you wanted to have a quad respawn window and randomize the first quad spawn, your powerupflags would be IF\_QUAD\_WINDOW + IF\_RANDOM\_FIRST\_QUAD = 1 + 4 = 5.

**6.6.35. levelstart [time\_seconds]**

Specifies the number of seconds (countdown) for the server to wait at the start of each level before the level actually begins. Defaults to 30 seconds. Note that almost no one gets into the game until 5-10 seconds have passed, so if you want your server to have a fair level countdown, this value should always be at least 15 seconds.

**6.6.36. levelstart\_uneven [time\_seconds]**

Specifies the number of seconds (countdown) for the server to wait at the start of a level if the previous level had uneven teams skill-wise. The teams are considered uneven if the difference in captures for the previous level was greater than or equal to half of the capturelimit. For example, if the capturelimit was 8, then the teams are considered uneven if the capture difference was 4 or more. Defaults to 45 seconds. Set this to "0" if you do not want the level countdown to consider uneven teams, in which case the countdown specified by the "levelstart" cvar will always be used instead.

**6.6.37. limitarenachat [0/1]**

During rocket arena overtime, controls whether global chat from eliminated players and spectators will not be heard by players that are still in the game. Defaults to ON (chat is limited). Limiting the global chat (say command) in this manner prevents eliminated players from spectator cheating during the rocket arena overtime. Spectator cheating includes giving out health numbers and player locations.

**6.6.38. logadmin [0/1]**

Controls whether attempts to gain administrator access are logged. Also controls whether the use of admin commands is logged. Defaults to ON. Changes to this cvar only take effect after the server is restarted.

**6.6.39. logbots [0/1]**

Controls whether bot detections are logged. Bot detections are written to the name log, so "lognames" must also be active in order to log bots. Defaults to ON.

**6.6.40. logchat [0/1]**

Controls whether chat messages are logged. Defaults to OFF. Changes to this cvar only take effect after the server is restarted.

**6.6.41. logfrags [0/1]**

Controls whether frag stats are logged. Defaults to OFF. Changes to this cvar only take effect after the server is restarted.

**6.6.42. lognames [0/1]**

Controls whether IP address and players names are logged. Defaults to OFF. Changes to this cvar only take effect after the server is restarted.

**6.6.43. maploop [0/1]**

Controls whether the MapLoop.cfg file is read after each level to determine the next map to load. Defaults to OFF.

**6.6.44. maploop\_file [file]**

Specifies the name of the maploop configuration file. This file must be located in your configuration directory. Defaults to "MapLoop.cfg".

**6.6.45. matchoptions [0/1]**

Controls whether match commands can be used. Defaults to OFF. Note that even if this cvar is ON, using any match command requires the appropriate administrator permissions.

**6.6.46. matchvote\_allow [0/1]**

Controls whether match admin elections are allowed. Defaults to OFF.

**6.6.47. matchvote\_minclients [0/1]**

Specifies the minimum number of clients that have to be connected to the server before a match admin election can take place. Defaults to “2”.

**6.6.48. matchvote\_pass [percentage]**

Specifies the minimum percentage of “yes” votes required to elect a match admin. Note that this is a percentage of “yes” votes out of all of the clients, not just the clients who voted. So all players who did not vote are essentially counted as “no” votes. Defaults to “51”.

**6.6.49. matchvote\_selfelectlimit [limit]**

Specifies the maximum number of times per level that a client can attempt to elect himself to a match admin. Clients are allowed to attempt to elect other admins an unlimited number of times. Defaults to “3”.

**6.6.50. matchvote\_time [time\_secs]**

Specifies the voting period in seconds. After the voting period is over, the votes are tallied and the results are posted. Defaults to “30”. Note that if the vote outcome can be determined before the voting period ends, then the voting period will terminate early. For example, if every client votes “yes”, then the voting period will end because there is no point in waiting through the entire voting period.

**6.6.51. maxping [max\_ping]**

Specifies the maximum ping a client is allowed to have to play in the game. A value of 0 turns off server checking for maximum ping. Defaults to 0.

**6.6.52. maxrate [rate]**

Specifies the maximum number of bytes that will be sent to each client per second. This is useful for preventing LPBs from sucking all of your server’s bandwidth. Defaults to 0 (no rate limiting).

**6.6.53. maxspawns [count]**

Specifies the maximum number of spawns per second during match or level starts. Limiting the number of spawns per second prevents clients from overflowing when they are spawned into the game after countdown. Defaults to 4, which allows 4 players per second per team. It is recommended that you do not change this cvar.

**6.6.54. maxteamchange [count]**

Specifies the maximum number of times a player may change teams during a single level. This restriction is only in effect if the “antiteamflood” cvar is ON. Defaults to “10”.

**6.6.55. mercylimit [0/1]**

Specifies the difference in captures that will prematurely end the level. This is blowout limit or mercy rule setting and is useful for avoiding 8-0 games where the teams are obviously uneven. Defaults to 0. A reasonable value for the mercylimit when the capturelimit is 8 might be 5 or 6.

**6.6.56. minping [min\_ping]**

Specifies the minimum ping a client is allowed to have to play in the game. A value of 0 turns off server checking for minimum ping. Defaults to 0.

**6.6.57. noclustertechspawn [0/1]**

Controls whether an improved start-of-level tech spawning algorithm is used that prevents such things as multiple techs clustering at a single spawn point. Defaults to ON.

Note that using this setting does not mean techs will be evenly spawned in the level so each team has an equal chance to get two techs each. This setting only prevents obviously bogus tech spawns in the form of multiple techs spawning at a single spawn point. It is still possible for all techs to spawn on one side of a level, giving one team an obvious advantage at the start of the level. Although this example would be rare, it can still happen because the tech spawn algorithm randomly spawns techs at player spawn points. It's still possible for all of the techs to spawn at different spawn points that all happen to be on the same side of the level.

#### **6.6.58. offhandhook [0/1/2]**

Controls whether the offhand grappling hook will be available. A value of "1" allows offhand hook and also allows players to continue to "use grapple". A value of "2" allows offhand hook but disables "use grapple". Default is 0 – no offhand hook available.

#### **6.6.59. ot\_rarounds [1/3]**

Specifies the number of rounds that will be played in rocket arena overtime. Defaults to 3 rounds. Settings this cvar to 0 or 1 will give you a single round. Setting this cvar to 2 or more will give you 3 rounds.

#### **6.6.60. ot\_sdtimelimit [time\_mins]**

Specifies the timelimit (in minutes) of the sudden death overtime. This timelimit is independent of the regulation timelimit (the "timelimit" cvar). Default is 10 minutes.

#### **6.6.61. ot\_style [0/1/2]**

Specifies the style of overtime used when the CTF game ends and the captures are tied. Defaults to 0. Set this to "1" for rocket arena overtime, and "2" for sudden death overtime.

#### **6.6.62. powerupflags [flags]**

Specifies what powerups you want to ban.

PF_BAN_QUAD	1
PF_BAN_INVUL	2
PF_BAN_SILENCER	4
PF_BAN_REBREATHER	8
PF_BAN_ENVIROSUIT	16

The powerupflags cvar works like the standard dmflags cvar: you pick the flags you want and add their values together to get the final powerupflags value. For example, if you wanted to ban the quad and the invulnerability, your powerupflags would be PF\_BAN\_QUAD + PF\_BAN\_INVUL = 1 + 2 = 3.

Changes to any of these ban flags only take effect the next time a level is started. Defaults to 0.

#### **6.6.63. qsgrapplefly [speed]**

Specifies the speed of the quad super grappling hook in flight – the speed of the hook before it hits a solid object and starts pulling the player. Defaults to 2000.

#### **6.6.64. qsgrapplepull [speed]**

Specifies the speed that the player is pulled by the quad super grapple. Defaults to 1500.

#### **6.6.65. savehighscores [0/1]**

Controls whether highscores are saved and displayed for each level. Defaults to OFF. Changes to this cvar only take effect after the server is restarted.

**6.6.66. savescores [0/1]**

Controls whether scores are saved for each level so players can use the “scores” command to see older scores. Note that this cvar has nothing to do with saving and displaying highscores. Allowing the use of “savescores” allows the user to type “scores <level>” and get a list of scores. Defaults to ON. Changes to this cvar only take effect after the server is restarted.

**6.6.67. sgrapplefly [speed]**

Specifies the speed of the super grappling hook in flight – the speed of the hook before it hits a solid object and starts pulling the player. Defaults to 1000.

**6.6.68. sgrapplepull [speed]**

Specifies the speed that the player is pulled by the super grapple. Defaults to 1000.

**6.6.69. spawnprotect [time\_sec]**

Specifies how long (in seconds) a player is respawn protected after being killed. Note that a player automatically loses respawn protection as soon as he picks up a weapon or picks up the enemy flag. This prevents abuses in some levels where powerful weapons are very near to respawn points. Defaults to OFF (0 seconds).

**6.6.70. stdlogname [log\_file]**

Specifies the name of the log file for the frag stats. This log will be saved to the “Logs” subdirectory of your L-Fire CTF configuration directory. Defaults to “StdLog.log”. The current date is automatically appended to the file’s base name if the “dailyfraglog” cvar is set to “1”.

**6.6.71. supergrapple [0/1]**

Controls whether the Keyser Grapple tech appears in the game. Defaults to OFF. Changes to this cvar only take effect the next time a level is started.

**6.6.72. teambalance [0/1]**

Controls the display of team balance messages and also prevents players from joining the larger team when the team’s are unbalanced. Defaults to ON.

**6.6.73. tempbantime [time\_sec]**

Specifies the number of seconds a player is banned from the server when using the “bootban” command. Defaults to 180 seconds (three minutes).

**6.6.74. vwep [0/1]**

Controls the use of VWep. Defaults to ON. Changes to this cvar only take effect after the server is restarted.

**6.6.75. weaponflags [flags]**

Specifies whether you want quick weapon switching, and what weapons to ban.

WF_QUICK_SWITCH	1
WF_BAN_SHOTGUN	2
WF_BAN_SUPERSHOTGUN	4
WF_BAN_MACHINEGUN	8
WF_BAN_CHAINGUN	16
WF_BAN_GRENADELAUNCHER	32
WF_BAN_ROCKETLAUNCHER	64
WF_BAN_HYPERBLASTER	128
WF_BAN_RAILGUN	256

WF_BAN_BFG	512
WF_BFG_SUICIDE	1024

The weaponflags cvar works like the standard dmflags cvar: you pick the flags you want and add their values together to get the final weaponflags value. For example, if you wanted quick weapon switching and you wanted to ban the BFG, your weaponflags would be WF\_QUICK\_SWITCH + WF\_BAN\_BFG = 1 + 512 = 513.

Changes to WF\_QUICK\_SWITCH and WF\_BFG\_SUICIDE take effect immediately. Changes to any of the ban flags take effect the next time a level is started. Defaults to 0.

#### **6.6.76. welcometime [time\_sec]**

Specifies how long (in seconds) the top 10 highscore list is displayed to the player when the player first enters a level. Default is 12 seconds. A value of 0 turns off the top 10 display.

#### **6.6.77. zbotname [0/1]**

Controls whether the server will reject connection attempts from players that have “[ZBot]” in their name. This does nothing to combat hacked ZBots, but it prevents newbies with unhacked ZBots from connecting to the server. Defaults to ON.

## **6.7. Configuration Files**

This is a list of the L-Fire CTF configuration files and a description of what each one does. The format for any of the user editable files is explained in the file itself.

#### **6.7.1. Admin.cfg**

Specifies the administration passwords and the permissions for each admin user. This file is read whenever a client attempts to gain administration access with the “admin” command.

#### **6.7.2. Banned.cfg**

Specifies the IP addresses and corresponding subnet masks for banned players. This file is only read upon server startup. Issue an “sv loadban” command from the server console to reload any changes while the server is running.

#### **6.7.3. ChangeMap.cfg**

Specifies the maps that can be specified with the “changemap” command. For now, it’s necessary to have this file to validate attempts to change the map. If an invalid map is specified, the server will boot all of the clients. This file is read whenever the “changemap” command is used.

#### **6.7.4. LHelp.txt**

This file contains the text that is displayed for the “LHelp” command. There is a limit on the number of characters that should be sent to a client, so only the first 1200 characters (approximately) of this file are actually displayed to the client. This file is read whenever the “LHelp” command is used.

#### **6.7.5. MapLoop.cfg**

Specifies the order of the maps when the server changes levels. This file is read after the end of each level.

### 6.7.6. MOTD.cfg

Specifies the message of the day text that appears when a client connects to the server. This file is only read upon server startup. Issue an “sv loadmotd” command from the server console to load this file while the server is running.

### 6.7.7. ScorePreload.cfg

This file specifies which temporary score files are loaded when the server is started. Temporary scores are saved in the TempScores subdirectory of your L-Fire CTF configuration directory. The scores for any maps that are listed in this file will be loaded when the server is started, and these scores will be available to clients when they use the “scores” command.

If you are running a particular map and the map is not listed in this file, then the map’s scores will not be preloaded when the server starts. This is harmless, but the scores for the map won’t be available to the “scores” command until the map is actually played and the level is finished.

The reason why the map names must be specified in this file is because no cross-platform code has been written so that the server can simply scan the TempScores directory and load all of the temporary score files that it finds. Eventually, this will be the case, and this file will no longer be needed.

### 6.7.8. Shutup.cfg

Specifies which clients will be automatically silenced when they connect to the server. This file is only read upon server startup. Issue an “sv loadshutup” command from the server console to load this file while the server is running.

### 6.7.9. TextFilter.cfg

Specifies the words and phrases that the text filter will match and not match when filtering player names and chat messages. This file is only read upon server startup. Issue an “sv loadtextfilter” command from the server console to load this file while the server is running.

## 6.8. Using Outlands II

Here’s some step-by-step instructions for changing your L-Fire CTF configuration to use the new Outlands II map that Id released at the beginning of July, 1998.

### 6.8.1. Install the New Map

You can download Outlands II at various places, such as <http://www.planetquake.com> and <http://www.captured.com>. Simply extract the map archive to your baseq2/Maps directory. If you don’t have a Maps directory in baseq2, then just create it.

Instead of placing the map in your baseq2/Maps directory, you can place it in your CTF/Maps directory. Do this if you want to keep your CTF maps organized within the CTF directory. The disadvantage of this method is that other mods you may run outside of the CTF directory will not have access to the Outlands II map if it is not either copied to the other mod’s Maps directory or copied to the baseq2/Maps directory.

### 6.8.2. Add Outlands II to Your Map Loop

In order to run this map with the other Id CTF maps, you will need to use the custom map loop feature of L-Fire CTF. This is because q2ctf3 (the Smelter) specifies q2ctf4 (Outlands I) as its next map, and not q2ctf4a (Outlands II). All you need to do is edit the MapLoop.cfg file and either add a new line `=q2ctf4a`, or replace the line `=q2ctf4` with `=q2ctf4a`. Do a replacement if you no longer

want to run the old Outlands, and add the new line if you still want to run the old Outlands and also the new Outlands (in this case, the map cycle would be six maps instead of five). Finally, put **set maploop 1** in your server's configuration file (usually server.cfg).

### 6.8.3. Setup the Score Preloading

In order to preload the temporary level scores for Outlands II, edit the ScorePreload.cfg file and add the line **=q2ctf4a**. It does not matter where you put this line. It just needs to be added so the server will know to preload the temporary level scores for this new level. Leave the **=q2ctf4** line in the file, unless you know you will definitely not run the older Outlands anymore.

### 6.8.4. Setup the Highscores

If you are using the highscore feature, then make a copy of the NewScores.txt file in your HighScores directory and call it q2ctf4a.txt. You only need to do this if you want to preset the highscores for Outlands II as 15 scores by "Player" with 10 points. If you want to start out with blank highscores for this level, then just skip this step.

### 6.8.5. Setup the ChangeMap Validation

Finally, add the line **=q2ctf4a** to your ChangeMap.cfg file. This file lists the maps that are valid for the changemap command. You need to add Outlands II to this file so admins can switch to this map using the changemap command.