



L-Fire CTF

Version 0.87

June 22, 1998

Author: Kai Kenner "L-Fire"

Table of Contents

1. Introduction	2
2. Getting Started	2
2.1. Archive Contents.....	3
2.2. Quick Start	3
2.3. Default L-Fire CTF Configuration	3
2.4. Your Server Configuration.....	3
3. Running L-Fire CTF	4
3.1. Message of the Day	4
3.2. Highscores	4
3.3. Weaponflags.....	5
3.4. IP Banning	5
3.5. Silencing Players.....	6
3.6. Logging	6
3.7. Using Custom Map Loops.....	7
3.8. Using Entity Files	7
3.9. Creating Admin Users	8
3.10. Restricting Player Pings	9
3.11. Reviewing Scores	10
4. Running Clan Matches	10
4.1. Preparation Stage	10
4.2. Countdown	11
4.3. Ending the Match	11
4.4. Server Feature Overrides.....	11
5. Upgrading from L-Fire CTF 0.85	11
5.1. Configuration Changes between 0.85 and 0.86.....	11
5.2. Moving from 0.85 to 0.86	12
6. Appendix	13
6.1. Client Commands.....	13
6.2. Clan Match Commands.....	14
6.3. Admin Commands	14
6.4. Server Commands	15
6.5. Server CVARS	16
6.6. Configuration Files	18

Revision History

Date	Editor	Description
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 adds useful features to 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 clan match support, 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 web site at <http://yyz.com/LFire> or email to LFire@rdsi.com.

Copyright/Permissions

L-Fire CTF is copyright © 1998, 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 archive 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 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

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

ctf/gamex86.dll	L-Fire CTF binary (Win32)
ctf/gamei386.so	L-Fire CTF binary (Linux)
ctf/LFireCfg/Ent	Directory for entity files
ctf/LFireCfg/HighScores	Directory for highscore files
ctf/LFireCfg/Logs	Directory for log files
ctf/LFireCfg/TempScores	Directory for temporary score files
ctf/LFireCfg/Ent/nobfg_q2ctf4.ent	Custom entity file for Outlands
ctf/LFireCfg/HighScores/NewScores.txt	Empty highscore file template
ctf/LFireCfg/HighScores/q2ctf1.txt	Empty highscore file for q2ctf1
(Highscore files included for q2ctf1-q2ctf5 and as2m1-as2m7)	
ctf/LFireCfg/LFire087.txt	Readme file for version 0.87
ctf/LFireCfg/LFireCTFManual087.pdf	This manual
ctf/LFireCfg/Admin.cfg	File for admin permissions
ctf/LFireCfg/Banned.cfg	File for IP banning
ctf/LFireCfg/ChangeMap.cfg	File for validating "changemap" command
ctf/LFireCfg/LHelp.txt	File for "LHelp" command
ctf/LFireCfg/MapLoop.cfg	File for defining the custom map loop
ctf/LFireCfg/MOTD.cfg	File for defining the message of the day
ctf/LFireCfg/ScorePreload.cfg	File for preloading level scores
ctf/LFireCfg/Shutup.cfg	File for silencing users

2.2. Quick Start

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

Other than the gamex86.dll or gamei386.so, 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 anti-spam, team balancing, spawn protection, 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, the highscore list 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 scores 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. Weaponflags

Quick weapon switching and weapons banning is 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, and no weapons are banned.

3.4. IP Banning

Clients can be banned from connecting to 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 examples on how to use IP banning:

3.4.1. 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

3.4.2. 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.

3.5. 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.

3.6. Logging

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

3.6.1. 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. 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 1st to the 15th of the month appear in an “01” file, and all connections from the 16th to the 31st of the month appear in a “16” file. This keeps things a little organized and prevents the log files from growing too big.

3.6.2. 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”.

3.6.3. 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”.

3.6.4. 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.

The name for the frag stats log is specified by the “stdlogname” cvar. By default, this cvar is set to “StdLog.log”. Unlike the IP/Name, chat, and admin logs, the frag stats log will keep growing in a single file until the server is shutdown. Presumably, if you’re keeping frag stats, then you’ll be saving the frag log to some other location and clearing it every once in a while, or you can change the “stdlogname” cvar to save frag stats to a new file.

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/>

3.6.5. 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.

3.7. 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.

3.8. 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.

3.8.1. 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.

3.8.2. 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.

3.9. 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 clan 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.

3.9.1. Permissions

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

- | | | |
|---|--------------------------|---------------------------|
| A | Change dmflags cvar | (setdmflags command) |
| B | Change weaponflags cvar | (setweaponflags command) |
| C | Change fraglimit cvar | (setfraglimit command) |
| D | Change capturelimit cvar | (setcapturelimit command) |
| E | Change timelimit cvar | (settimelimit command) |
| F | Change antis spam cvar | (setantis spam 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) |

- P Allow use of clan 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 Ignore ping restrictions

3.9.2. 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 clan admin permission and allow them to use the forceteam command, the permissions string would be "PS" - the letter "P" gives the user clan admin permission, 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 the letters A through J or P through U are simply ignored.

Here are some example lines for admin users:

3.9.3. 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 A through J and P through U are listed.

```
=superuser mypassword/ ABCDEFGHIJPQRSTU
```

3.9.4. Clan Administrator

A clan match administrator needs permission to use the clan match commands. In this example, the username/password is just a single word "clanpassword321". The clan administrator would type "admin clanpassword321" at the console to gain permission to use clan match commands. The permission flag "P" is for clan match administration.

```
=clanpassword321/ P
```

3.9.5. 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 "U" allows a player to ignore ping restrictions.

```
=goodhpb/ U
```

3.10. 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.

3.11. Reviewing Scores

In standard CTF, once the server has changed levels, there is no way for the players to view the scores for the previous level. In L-Fire CTF, players can view the most recent scores for any level by typing "scores <level>". For example, to view the scores for the most recent time q2ctf1 was played, type "scores q2ctf1". Each time a level is played, it overwrites the scores for the previous time the level was played. The score saving feature is especially useful for clan matches, where the players can review the scores after the level has ended and the server has changed to the next level.

Note:

Scores for each level are only saved if the "savescores" cvar is set to "1". By default, this cvar is set to "1". If you do not want to save scores between levels, then set this cvar to "0".

4. Running Clan Matches

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

Five commands are used to manage a clan match: matchprepare, matchstart, matchend, matchstate, and changemap. In order to use clan 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 clan match commands.

4.1. Preparation Stage

To start a clan match, use the "matchprepare" command. Once this command is given, the server switches to clan match mode and the server runs in the clan match preparation 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 preparation 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.

4.2. 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.

4.3. Ending the Match

Whenever the server changes levels, either due to an explicit map change or when the current level ends, the server will drop out of clan match mode and switch to non-match mode. To begin a new match, issue a “matchprepare” command.

If the server is currently in the clan match preparation or countdown stage and there are no clan administrators connected (if, for example, the clan administrator disconnected), then the server automatically switches to non-match mode. This is done to prevent players from being stuck as spectators. 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 clan administrators connected. This is done so server overrides stay in effect during the match (see the section below on server overrides).

4.4. Server Feature Overrides

In order to prevent non-clan match features from interfering with a match, the server automatically disables anti-spam, team balancing, and ping checking. In addition, the reconnect resume option is automatically enabled. When the match is ended, the anti-spam, team balancing, ping checking, and reconnect resume options are restored to their non-match values.

5. Upgrading from L-Fire CTF 0.85

Several changes have been made between L-Fire CTF version 0.85 and L-Fire CTF version 0.86. It is very important to read the following upgrade instructions if you want to maintain your previous L-Fire CTF highscores and other configuration files.

5.1. Configuration Changes between 0.85 and 0.86

1. The files Admin.cfg, ChangeMap.cfg, MapLoop.cfg, and ScorePreload.cfg were added to the L-Fire CTF configuration directory.
2. The message of the day file has been renamed from MOTD.txt to MOTD.cfg.
3. The subdirectories Ent, Logs, and TempScores were added as subdirectories in the L-Fire CTF configuration directory.
4. The highscore subdirectory has been renamed from Scores to HighScores.
5. The file nobfg_q2ctf4.ent is included in the Ent subdirectory.
6. The format of the Banned.cfg file changed. The previous version assumed that any line

beginning with a pound sign (#) was a comment, and all other lines had an IP address and subnet mask. The new version assumes any line is a comment, unless it begins with an equal sign (=). This makes the Banned.cfg file more consistent with the other configuration files.

7. The highscores cvar has been renamed to savehighscores, and the default for the cvar has changed from 1 to 0. The cvar was renamed because on a single player test game, the “highscores” cvar conflicted with the “highscores” command. The default was changed from 1 to 0 to make it easier to setup L-Fire CTF. Server operators can start using the various features one at a time, instead of having them activated by default.
8. The default for the weaponflags cvar has changed from 513 (quick weapons and ban BFG) to 0 (standard switching and no weapons banning).
9. The email and website serverinfo cvars were removed. They were included in the previous versions by accident.
10. The antispam, spawnprotect, vwep, vweprestrict, and idledetect cvars were removed from the public serverinfo list. These variables will no longer show up in game server query programs like GameSpy. The default values for these cvars were not changed. These cvars were removed from the serverinfo list because they don’t seem important enough to be included, and the serverinfo list was getting too big, which can cause an overflow problem when the server is queried.

5.2. Moving from 0.85 to 0.86

Here’s a step-by-step list of instructions that will help you move your version 0.85 installation to 0.86.

1. Remember that file and directory names are specified in this document in mixed case. If you are using Linux, then all file and directory names must be in lower case.
2. Replace the gamex86.dll or gamei386.so file in your ctf directory with the new version.
3. Add the Admin.cfg, ChangeMap.cfg, MapLoop.cfg, and ScorePreload.cfg files to your L-Fire CTF configuration directory.
4. Create the directories Ent, Logs, and TempScores in your L-Fire CTF configuration directory.
5. Place the nobfg_q2ctf4.ent file in the Ent directory. If you want to use the custom entity placement for the Outlands, then rename “nobfg_q2ctf4.ent” to “q2ctf4.ent”.
6. Rename the Scores directory to HighScores.
7. Rename your MOTD.txt file to MOTD.cfg.
8. Edit your Banned.cfg file and place an equal sign (=) at the beginning of every IP/subnet line.
9. If you were using highscores, then you must place “set savehighscores 1” in your server’s configuration file (usually server.cfg). This will take care of the cvar being renamed from “highscores” to “savehighscores”, and it will also handle the default being changed from ON to OFF.

10. If you were using the default weaponflags value of 513 (quick weapon switching and ban BFG), then you must place “set weaponflags 513” in your server’s configuration file. If you were not using the default weaponflags, then you most likely already have a “set weaponflags” line in your server’s configuration file. In this case, you can leave the “set weaponflags” line alone.

6. Appendix

6.1. Client Commands

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

6.1.1. 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.

6.1.2. highscores [level]

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

6.1.3. ignore [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.

6.1.4. lhelp (LHelp, info)

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

6.1.5. 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.

6.1.6. spectator (observe, observer)

Changes the player to a spectator.

6.1.7. 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.

6.1.8. time

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

6.1.9. unignore [player_number]

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

6.2. Clan Match Commands

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

6.2.1. 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.

6.2.2. matchend

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

6.2.3. matchprepare

This command is used to put the server into clan match preparation mode. The server will be in the “preparation” 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.

6.2.4. matchstart [time_secs]

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

6.2.5. matchstate

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

6.3. Admin Commands

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

6.3.1. admin [password]

This command is used by a client who wants to gain administration access. Typing “admin” without a password will display a summary of the admin permissions that have been granted to the client.

6.3.2. boot [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.

6.3.3. changemap [map]

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

6.3.4. forceteam [player_number] [red/blue/spec]

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

6.3.5. setantispam [value]

Displays and sets the antispam cvar.

6.3.6. setcapturelimit [value]

Displays and sets the capturelimit cvar.

6.3.7. setdmflags [flags]

Displays and sets the dmflags cvar.

6.3.8. setfraglimit [value]

Displays and sets the fraglimit cvar.

6.3.9. setmaxping [value]

Displays and sets the maxping cvar.

6.3.10. setminping [value]

Displays and sets the minping cvar.

6.3.11. setspawnprotect [value]

Displays and sets the spawnprotect cvar.

6.3.12. setteambalance [value]

Displays and sets the teambalance cvar.

6.3.13. settimelimit [value]

Displays and sets the timelimit cvar.

6.3.14. setweaponflags [flags]

Displays and sets the weaponflags cvar.

6.3.15. shutup [player_number]

This command allows an admin user to silence another player by specifying the player number (same number as displayed by “status” and used in the “kick” command). A client without rcon can get the player number by typing “players”.

6.3.16. unshutup [player_number]

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

6.4. 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.

6.4.1. 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.

6.4.2. sv loadban

Reloads the IP banning file (Banned.cfg).

6.4.3. sv loadmotd

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

6.4.4. sv loadshutup

Reloads the player shutup file (Shutup.cfg).

6.4.5. sv shutup [player_number]

Same as the admin “shutup” command.

6.4.6. sv unshutup [player_number]

Same as the admin “unshutup” command.

6.5. Server CVARS

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

6.5.1. allowresume [0/1]

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

6.5.2. antispam [0/1]

Controls anti-spam checking. Defaults to ON.

6.5.3. 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.

6.5.4. clanoptions [0/1]

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

6.5.5. configdir [dir]

Specifies the root of the configuration files. This directory must be a subdirectory of your quake2/ctf directory, for example, c:\Quake2\ctf\LFireCfg. Defaults to “LFireCfg”.

6.5.6. consolesay [0/1]

Controls whether non-commands typed directly in the console are treated as chat messages. Defaults to OFF. It's recommend 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.

6.5.7. 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.

6.5.8. 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.

6.5.9. 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.

6.5.10. 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.

6.5.11. logchat [0/1]

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

6.5.12. logfrags [0/1]

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

6.5.13. 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.

6.5.14. maploop [0/1]

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

6.5.15. 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.

6.5.16. 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.

6.5.17. 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.

6.5.18. spawnprotect [time_sec]

Specifies how long (in seconds) a player is protected after he respawns after being killed. Default is 2 seconds. A value of 0 turns off spawn protection.

6.5.19. 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”.

6.5.20. 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.

6.5.21. teambalance [0/1]

Controls the display of team balance messages. Defaults to ON.

6.5.22. vwep [0/1]

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

6.5.23. vweprestrict [0/1/2]

Controls whether vwep player or weapon models are restricted to male, female, and cyborg. Using vweprestrict will prevent white cubes from appearing on players if the client does not have the correct model on his computer. Set this cvar to “1” to enable player model restriction. Set it to “2” to enable weapon model restriction. Defaults to “1”.

6.5.24. 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.

6.5.25. 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. 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.

6.6.1. 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.

6.6.2. 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.

6.6.3. 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 kick off all of the clients. This file is read whenever the “changemap” command is used.

6.6.4. 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.

6.6.5. MapLoop.cfg

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

6.6.6. 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.

6.6.7. 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.

6.6.8. 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.