

L-Fire Release Notes

Version 2.0 Beta 125

Author: Kai Kenner

Table of Contents

1. L-Fire 2.0 Beta 125 Release Notes	1
2. Upgrading From L-Fire CTF/DM 1.x	1
3. Changes and New Features	3
3.1. Changes and New Features for both L-Fire CTF and L-Fire DM	3
3.2. Changes and New Features for L-Fire CTF Only.....	7
3.3. Changes and New Features for L-Fire DM Only.....	9
4. Bug Fixes	9
4.1. Bug Fixes to both L-Fire CTF and L-Fire DM	9
4.2. Bug Fixes Specific to L-Fire CTF	10
4.3. Bug Fixes to id Source Code	10
5. Special Features	11
5.1. Improved Match Mode	11
5.2. New Admin Permissions	12
5.3. Intermission Map Voting.....	13
5.4. Priority Connect.....	14
5.5. L-Fire CTF and L-Fire DM integration	16
5.6. Map-Specific Config Files	16
6. New CVars	19
6.1. New CVars for both L-Fire CTF and L-Fire DM.....	19
6.2. New CVars for L-Fire CTF Only	21
6.3. New CVars for L-Fire DM Only.....	22
7. New Commands	22
7.1. New Client Commands	22
7.2. New Server Commands	23
8. Upgrading From Previous L-Fire 2.0 Betas	24
8.1. Important Upgrade Instructions (2.0b110 and 2.0b115)	24
8.2. Changes and New Features for Both L-Fire CTF and L-Fire DM.....	24
8.3. Changes and New Features for L-Fire CTF Only.....	26
8.4. Bug Fixes	27

1. L-Fire 2.0 Beta 125 Release Notes

Please READ these release notes! Yes they're long but they're written for your benefit. These notes assume that you are an experienced L-Fire CTF/DM server admin. Please do not run this beta if you do not have experience running an L-Fire server.

Upgrading from a Previous L-Fire 2.0 Beta

If you are currently running an L-Fire 2.0 beta server (2.0b110 or 2.0b115), then you can skip most of this document and simply read the short section 8 at the end of this document - "Upgrading from Previous L-Fire 2.0 Betas".

Warning

This is a beta release and may be extremely unstable. Expect several hard server crashes per day. Also be aware that commands, cvars, and functionality may change before the final release.

Bug Reports

Please email bug reports to lfire-bugs@yyz.com or you can post bug reports to the L-Fire message board: <http://yyz.com/cgi-bin/lfire/Ultimate.cgi>

Support

Please DO NOT email anyone on the L-Fire team regarding 2.0 beta support questions as we're all swamped with mail and work, and unfortunately we haven't been able to answer all the mail that comes in. You'll have a much better chance of getting your question answered in a timely manner if you post to the L-Fire message board: <http://yyz.com/cgi-bin/lfire/Ultimate.cgi>

Thanks!

2. Upgrading From L-Fire CTF/DM 1.x

This section is intended for server admins who are currently running an L-Fire 1.x server. If you are already running an L-Fire 2.0 beta server, then you can skip to section 8 – "Upgrading from Previous L-Fire 2.0 Betas".

Version 2.0 has a new way of managing server configuration info (motd, admin permissions, banned addresses, etc.) that is a lot cleaner than the 1.x versions. Instead of having several .cfg files in the LFireCfg directory, 2.x does away with all that and puts all of your config information in a single file (lfire.ini) that is located in the mod directory (baseq2, lfiredm, ctf, etc.) All score information (e.g. highscores) is put in a single file such as highscores.ini that is also located in the mod directory.

Follow these step-by-step instructions to upgrade your server:

- 1) To make it as easy as possible for server admins to upgrade, version 2.0 is fully backwards compatible with the 1.x config file scheme and 1.x cvars, so the first thing to know is that you don't really have to do *anything* to run 2.0. You can simply replace the L-Fire 1.x gamex86.dll and restart the server. L-Fire 2.0 will first attempt to find config information in the new format, and if it can't find the new lfire.ini config file, it'll fall back to the old LFireCfg scheme.
- 2) If you're saving highscores:

As the server loads levels, it'll automatically migrate highscores for each level from

LFireCfg/Highscores/<level>.txt to the highscores.ini file. When you first start the server, there won't be a highscores.ini file in your mod directory. Don't worry about this – the server will create the file for you when it sees that you don't have one.

Since the server only migrates one level at a time as each level is loaded, you must let the server run a full map cycle and load all of the levels before every level's highscores will be in the new format. You can then safely remove your old highscore files.

You can speed the migration process by connecting to the server and typing at the client console "highscores <level>". The server will display the highscores for each level you specify; and as those highscores are loaded, 2.0 will automatically save the highscores in the new format.

Note that the server now saves the top 50 highscores for each level, but only displays 15 at the end of the level. In addition, it'll display 20 if you use the "highscores <level>" command. So if you migrate your highscores using "highscores <level>", you'll notice that the last 5 scores displayed on your console will be 0, since the 1.x versions didn't save more than 15 scores.

- 3) When you're ready to fully migrate from the old 1.x config file scheme to the 2.0 lfire.ini scheme, rename the file lfirectf_sample.ini (for CTF) or lfiredm_sample.ini (for DM) to lfire.ini and then edit lfire.ini. Editing instructions can be found inside the ini file itself. You'll have to import most of your configuration information from the old config files into the new lfire.ini. Note that some of the sections within lfire.ini don't have the same format as the old config files (notably the motd section). Also be aware that the admin permissions have changed between L-Fire 1.x and L-Fire 2.0, so you will have to redo your admin permissions when you migrate to the lfire.ini configuration scheme.

Important

It is highly recommended that you migrate your configuration information from the old 1.x scheme to the new 2.x lfire.ini scheme as soon as possible. This is because there is a security bug in Quake2 that allows clients to download your config files if your server has allow_download turned on.

Until you've had a chance to migrate your configuration information to lfire.ini, you can close this security loophole by turning off autodownload (set allow_download 0). Once you migrate your configuration, you can turn autodownload back on, and then you should rename your LFireCfg directory to something obscure so clients can't guess the directory name.

- 4) If you're using any custom entity files, you should move your Ent directory to your mod directory. For example, if the current Ent directory is quake2/ctf/lfirectf/ent, you should move it to quake2/ctf/ent.
- 5) There's a new data file called mapnames.ini that must be placed in the game directory (baseq2, lfiredm, ctf, etc.) This file lists maps and their titles (q2dm1-The Edge, lfctf1-Neurotica, etc.) and this file is where L-Fire 2.0 gets map titles to be displayed in the optional intermission voting menu.
- 6) Finally, because of the same Quake2 security bug described above, logfiles are not saved in the mod directory. By default they're saved in a subdirectory of quake2. The log subdirectory is named after the mod directory with "_logs" appended. For example, if the mod directory is baseq2, the server will save logs in quake2/baseq2_logs. For CTF, the server saves logs in quake2/ctf_logs.

You do not have to create the log directory yourself. The server will create it for you if it doesn't find one.

3. Changes and New Features

3.1. Changes and New Features for both L-Fire CTF and L-Fire DM

- **Priority connect.**
Tired of not being able to connect to your own server because it's always full? Would you like to give preferred players a password so they have higher priority in connecting to your busy server than a non-preferred anonymous player? Well priority connect is here and this new feature kicks serious ass! See detailed instructions later in this document.
- Intermission map voting. This is another really cool feature in L-Fire 2.0. Highly recommended. See details later in this document.
- Match mode has been significantly enhanced. See details later in this document.
- L-Fire CTF and L-Fire DM integrated into a single mod. See details later in this document.
- Custom config files can now be executed for each map, so map-specific settings are now possible. For example, you may want to disable the BFG on some maps but not others, or you may want to specify rocket arena overtime for some maps and sudden death overtime for others. See details later in this document.
- Most server cvars are now totally dynamic - they can be changed at any time and the changes will take effect immediately. For example, in L-Fire 1.x when you change the weaponflags to ban the BFG, the BFG only disappears when you restart the level. With L-Fire 2.x the change will take effect immediately – you'll see the BFG disappear and the BFG will be removed from player inventories.
- Added in-eyes chase camera. Thanks to David “crt” Wright for the code.

The chase/eye camera can be selected from the Join menu and then you can toggle between the chase/eye camera modes when you're a spectator by pressing the FIRE button.

- Duplicate maps are now allowed in the map list so very long map loops can be specified. The relative frequency of maps can be adjusted by using some maps in the list more often than others. For example, you may want q2dm1 to play more often than q2dm2. In this case, make a very long maplist and have q2dm1 appear more often in the list than q2dm2.
- Random map option. This option tells the server to pick maps from the map list at random. You can adjust the probability that a map will be picked by listing the map more or less frequently in the list than other maps. For example, if you put 15 different maps in the map list and then add q2dm1 to the list and repeat it 5 times, on average, q2dm1 will be picked about 25% of the time.

The server does not blindly pick maps out of the list. Instead, it won't pick the five previous maps that were played. For example, if the server had just run q2dm1, q2dm2, q2dm3, q2dm4, and q2dm5, the server won't pick any of those maps for the next map since they are within the

last five maps played. Note that the server may be forced to violate this rule if you don't have enough choices in the map list (if you only put two maps in the map list, for example).

- Server map selection based on number of players. For each map in the map list, you can specify an optional minimum and maximum number of players for the map. If the minimum or maximum restrictions are not met, then the server will skip that map.

Note that the server does not consider spectators in the total player count when checking range limits.

- Support for id 3.20 "sv_maplist" cvar. The sv_maplist is like the old L-Fire 1.x maploop. It's only implemented in L-Fire 2.x so that L-Fire 2.x will be admin-compatible with id's DM and CTF mods. Unless you're defining very simple maplists, it's recommended that you use the L-Fire 2.x maplist feature as it is much more powerful and flexible than using sv_maplist.

Note that sv_maplist overrides the maplist in lfire.ini. By default sv_maplist is blank. If you define a maplist in sv_maplist, it will always be used instead of the one in lfire.ini (or the legacy LFireCfg/maploop.cfg). This is just for the convenience of new L-Fire 2.x server admins.

- Map voting. Clients can type "mapvote <level>" to start a vote for a new level. Other players on the server can type "vote yes" or "vote no" to accept or reject the proposed map. If a player does not vote, then his vote is counted as "no".
- Added player forwarding. This feature is used when you're moving your server to a new IP address or port and you want the server at the previous address to automatically forward all connecting clients to the new server.

To use this feature, set the "forward_address" cvar to the new address, such as "192.168.0.1:27910", and then restart the server. When players connect to your server they will see the message of the day and will be automatically reconnected to the new address when they try to advance past the message of the day (by pressing ENTER, TAB, ESCAPE, etc.)

You should put some text in your message of the day like this: "SuperAwesome DM has moved to 192.168.0.1 (quake.superawesome.com). Please update your address books with the new server address."

When the server stuffs a connect command to the player, the player will also get the following message: "You are being forwarded to <address>". <address> is the address specified in the forward_address cvar. You can replace <address> with your own text such as "192.168.0.1 (quake.superawesome.com)". Then the player will get a message like this: "You are being forwarded to 192.168.0.1 (quake.superawesome.com)".

- Newlines are filtered from chat text so mischievous players can't do "anonymous" chatting.
- Added forcejoin cvar. Setting forcejoin to 1 will cause players who connect to the server to immediately join the game. In DM, players will be spawned into the level. In CTF, players will be given a team and spawned into their base.
- Added killer's average ping statistic to stdlog frag output.
- Added support for info_position entities (mostly useful for CTF or Team DM). These are entities that contain location information for the map so instead of the %L chat macro expanding

to a string like “by the red shotgun”, it’ll expand to a string like “near the red railroom”.

- Added player shutup by IP address. This is supported in the [ipshutup] section of the lfire.ini file. The IP address format is the same as in the [ipfilter] section. Note that the “sv loadshutup” command will now reload both the [shutup] and [ipshutup] sections of the lfire.ini file.
- Added support for adding additional entities to a map instead of having to do a full replacement of the bsp’s entities. In L-Fire 1.x, a <level>.ent file would replace all of the bsp’s entities with the entities specified in the ent file. Now you can put #add at the beginning of your ent file and this will instruct the server to add the file’s ents to the ones in the bsp, and not do a full replacement. So if all you wanted to do is add a power shield to a map, your ent file would look something like this:

```
#add
{
  "origin" "700 200 -400"
  "classname" "item_power_shield"
}
```

- You can clear the password, priority_password, and superuser_password cvars by setting them to "none".
- Changed the player booting code because there's a new client proxy called "Zorbot" that prevents players from being booted in previous Fire versions unless the console "kick" command is used.
- Added new ZBot detection code. The Zorbot proxy prevents older L-Fire versions from detecting ZBots. The new detection code detects ZBots even if the Zorbot proxy is used.

Important

If you suspect false positives with this new ZBot detection code, please notify me immediately!

- id chat flood control is automatically turned off in match mode.
- The settings commands (psettings/msettings in L-Fire 2.0) now display what the standard value is supposed to be in parenthesis if a server setting is non-standard.
- Using any of the "set <variable>" commands without any argument will display the cvar value with a list of valid values.
- The id "gamedate" cvar was removed. The gamedate cvar was an id serverinfo variable that contained the build date of the mod. There's no need to have this taking up valuable serverinfo cvar space when the version info is in the L-Fire CTF/DM version string.
- Server displays player counts whenever a player connects, disconnects, changes teams, or changes to observer.
- Added "playerlist" command, which lists all of the players, whether they are observer or playing, and whom the player is chasing (if any).
- Added "chaseme" command, which can be used by a player to list everyone who is currently chasing that player.

- Added "autodemo" command. This command causes the server to automatically start and stop client demos whenever the level changes. Each demo filename is formed from the level and date/time, such as q2dm1_990408_142304.dm2, which means the demo was started on q2dm1 on April 8, 1999 at 2:23:04 pm.
- The server now saves the top 50 highscores for each level. It will display the top 15 at the end of the level, and it will list the top 20 if you use the "highscores <level>" command.
- Support for id 3.20 IP filtering: sv addip, sv removeip, sv listip, sv writeip, filterban. The L-Fire method for IP filtering (the banned.cfg file) still works. The server now checks the id IP filter list and then checks the L-Fire IP filter list. A match in either list will result in the client's connection attempt being rejected.

Note that the id "filterban" cvar reverses the meaning of the IP filter lists. By default, filterban is set to "1", which means that a match in either list results in the client being banned (allow all except those that match either list). By setting filterban to "0", clients that don't match at least one of the filter lists will be banned. So the filter lists become a "deny all except those that match either list".

Also note that the banned.cfg is no longer called the "banned" list since the filterban cvar can reverse its meaning. In the lfire.ini file, the IP addresses are listed under an [ipfilter] section, which is a more accurate term.

- Support for id 3.20 flood control cvars: flood_msgs, flood_persecond, and flood_waitdelay. By default, id antiflood is turned off because the L-Fire flood control (antispam) is more sophisticated. You can use id antiflood and L-Fire antispam at the same time, although it's not necessary.
- Removed nearly all of the single player game code (monster code, co-op support, etc.) This change is transparent to players and admins, but it makes the code a lot cleaner and a tiny bit faster.
- The player ID "Viewing..." string no longer includes the player's model and skin.
- Using the "kill" command does not immediately respawn players. Instead, the player is respawned after a second (if force respawn is on), or after the player presses the fire button (if force respawn is off). This is an id 3.20 code change.
- The server now lists valid maps if the "changemap" command is used without specifying a map or if an invalid map is specified.
- If the server password is not set, or it is set to the string "none", the client password is not checked. So a client can now connect to the server even if he has set a client password, as long as the server itself hasn't set a password.
- If a client is in spectator mode, the text "SPECTATOR MODE" will appear in the lower left of the screen. This is particularly helpful to newbie players who wonder "why can't I shoot anyone and how come I can walk through walls?"
- If a client is chasing another player, the text "Chasing <player>" will appear immediately above the "SPECTATOR MODE" text.

- When chasing a player who dies, the chaser's view is tilted, just as the player's view is tilted (from id 3.20 integration).

3.2. Changes and New Features for L-Fire CTF Only

- Scoreboard now indicates "Match Mode" if the server is in match mode.
- Added team chat HUD. This is an experimental feature to help CTF teamplay. One of the problems with say_team chat text is that many times your teammates' chat text scrolls off the console before you've had a chance to read it. This is especially true if the chat text comes when you're busy fighting or a lot is happening and obituary messages are coming quickly. The chat HUD aims to solve this problem by saving a 30 second history of chat messages. The messages are displayed in the lower left corner of the players' screens. For players with at least 800x600 screen resolution, the chat text doesn't intrude much on the HUD. For people playing with a lower resolution, a key can be bound to toggle the chat HUD only when the player wants to see the team chat buffer.

The advantage of the chat HUD over bringing down the console to see old chat text is obvious. Instead of being vulnerable as half your screen is covered by the console and not being able to move either, you can move around and read the chat text at the same time. The disadvantage of the chat HUD is that you really need a minimum of 800x600 resolution so it doesn't intrude too much on the player's screen. At 1024x768 resolution or above, the chat HUD is almost a no-brainer as it hardly intrudes on the player's screen at all.

Toggle the chat HUD by using the "chathud" command. By default, the chathud is 50 columns by 8 rows and the history is 30 seconds. At this time the chat HUD is not player configurable, but it will be in a future release. For example, players running at 1024x768 or above might want a 80x10 display with a 60 second history.

Depending on player feedback during beta testing, this feature may or may not be included in the final L-Fire 2.0 release.

- Forcing a player to spectator no longer triggers the team change flooding detection. This was a problem for admins as they could only force players to a team/spectator a limited number of times before the team change flood detection kicked in and prevented further forceteam commands on the player.
- Player scores are now reset to zero at the beginning of each rocket arena overtime round, making it much easier to tell who did well during the round.
- Removed the colorquad cvar. This option is no longer necessary as colored quad (red/blue glow depending on team) is now standard across all CTF mods.
- Removed the noclustertechspawn cvar. This variable was used to control whether the improved L-Fire tech spawn algorithm was used or the older id CTF tech spawn algorithm was used. The problem with the id algorithm was that it allowed multiple techs to spawn at the same spawn point, and some spawn points were favored over others. The id CTF tech spawn algorithm was removed in the L-Fire 2.0 code cleanup so there's no need for this cvar anymore.
- Added %X chat macro which expands to the location of all techs that are in the player's line of sight. For example, the text "Tech spotted: %X" may expand to "Tech spotted: Autodoc near the

red flagroom, Power Amp in the red main entrance”. Note that the location is that of the tech, not the player sending the chat message.

- Added %C chat macro which expands to the location of all flag carriers that are in the player's line of sight. For example, the tech “%C” may expand to “Friendly flag carrier spotted in the water near the red water tube”. Note that the location is that of the flag carrier, not the player sending the chat message.
- Added %F chat macro which expands to "a Red Flag" or "a Blue Flag" as appropriate if the chatting player is currently carrying the red/blue flag.
- Added %Q chat macro which expands to "Quad" if the chatting player currently has quad.
- Server displays the total time a player held the enemy flag before capturing. Useful for "fastest capture" competitions.
- Ping rating for each team is displayed on the scoreboard. Ping rating for each team is calculated as the geometric mean of the players' average pings. That's why the scoreboard says "Ping Rating" and not "Average Ping".
- Added Rocket Arena intermission and RailWar intermission.
- The player count in RA OT (and RA/RW intermission) now shows the full count over 9 players, instead of just showing "+9 players".
- If a client in overtime or intermission has been eliminated, the text “ELIMINATED” will appear in the lower left of the screen.
- Free team change feature. If you set the "freeteamchange" cvar to 1, players are allowed to change teams for "free", which means they don't lose any of the items, health, armor, techs, etc. that they are carrying. Essentially they are just teleported to their new team's base and their skin is changed. They will lose their score though. This prevents players from changing teams to run up a highscore.

The purpose of this feature is to help with team balancing. Good players that are tanked up with a good tech may be more inclined to change to the losing team if they know they won't lose any of their items. Free team change works even if you use the forceteam command. So forcing a player to another team won't cause that player to lose any items (just their score).

- Added "monitor" admin command which allows spectators to see messagemode2 chat from the red and blue teams. Using the monitor command requires that the client have "shutup" admin permission. In addition, the client will only see both red and blue chats if he is in spectator mode. This prevents any possibility of cheating – once a monitor starts playing, he can't see the other team's messagemode2 chat anymore. If the player goes back to spectator, he can see red/blue messagemode2 chat again.

The monitor command also allows the monitor to see chat text from players who have been shutup, whether the shutup player is red, blue, or a spectator.

Finally, the monitor feature is totally disabled during match mode. Even if a spectator has activated the chat monitor, he won't see red/blue messagemode2 chat if the server is in match mode. Once again, this is to prevent any possibility of cheating.

- Full support for the Team 3 WTF CTF maps. Players now get credit for picking up and capturing multiple flags, among other things.
- Option to reset uneven teams at the start of a level. If the previous level ended with a blowout, then all of the players are moved to spectator when the next level begins. Players must then join teams manually. A blowout is defined as hitting the mercylimit or if the capture difference was more than half the capturelimit. For example, if the capturelimit was 10, then a blowout would be a capture difference of 6 or more.

Your mileage may vary on this one. In testing on YYZ-1, players tended to stay as spectators too much, instead of rejoining a team.

3.3. Changes and New Features for L-Fire DM Only

- Added fastest frags list option. The all-time fastest frags list saves the names and playing time of the top 50 players who hit the fraglimit the fastest. As with highscores, the server saves a fastest frags list for each level. Saving fastest frags instead of highscores is a lot more useful on DM servers where the server has a fraglimit. Instead of the highscore lists being populated by players that have all hit the fraglimit, the fastest frags list will recognize the players that hit the fraglimit in the least amount of time. Thanks to UV-Dredd for the suggestion.

If the fastest frags option is enabled, then highscores are automatically disabled.

Note: Your fastest frags list for each level will fill up very slowly because there is usually only one player hitting the fraglimit per level. So it typically requires that a map be played 15 times before all 15 slots on the end of level list are filled.

- Added "monitor" command which allows spectators to see chat text from players who have been shutup.
- Observers are no longer displayed on the scoreboard with a -999 score. Instead they are displayed with a score of 0 and the text "SPEC" is used to indicate that they are observers.
- The gamename cvar is now set to "baseq2" and a new "gameversion" cvar has been added which has the mod name and version, such as "L-Fire DM v2.0". This change was made so GameSpy filters and the CLQ would not filter out L-Fire DM servers.
- One thing to be aware of, L-Fire DM does NOT support the id method of moving to/from observer with the spectator 0/1 cvar. This is part of the id 3.20 code base, but the way it's implemented is really kludgy and it interferes with things like idle detection, so we decided not to support the client spectator cvar. As usual, clients can move to spectator by typing "observer", and they can join the game by hitting TAB and selecting "Join Game".

4. Bug Fixes

4.1. Bug Fixes to both L-Fire CTF and L-Fire DM

- Uppercase map names on linux would pass the changemap validation, but would crash the server

since filenames on Linux are case-sensitive.

- In demo playback, weapon pickups would sometimes have the "whoosh" level start sound instead of the normal weapon pickup sound.
- An erroneous "GameEnd 0.0" was being written to the frag logs each time the server was started. This apparently caused problems with some fragstats parsers.
- Server could quit with "SZ_Getspace: overflow without overflow set" in rare cases when a client was connecting and the server erroneously detected name change flooding from the connecting player.

4.2. Bug Fixes Specific to L-Fire CTF

- In demo playback, messages such as "can't find spawn point to spawn item_tech1" would be seen on the player's console. These messages have been eliminated.
- In RA overtime, you could use the "chase" command to chase a player who was already eliminated.

4.3. Bug Fixes to id Source Code

- Fixed an id CTF bug where the %L chat macro would indicate "in the water" if the player was in water, slime, or lava. The fixed code now correctly says "in the water", "in the slime", or "in the lava" as appropriate.
- Fixed an id bug where weapons in a loop don't rotate if weapons stay is on. For example, in Tokay's Towers there is a railgun and BFG that are supposed to randomly rotate when the weapon is picked up. When weapons stay is on, only the railgun appears. In L-Fire 2.0, the weapon does not stay when picked up, and a random weapon is chosen when the weapon respawns.
- If a dropped quad is picked up immediately before the quad is about to expire, the player would get a full 30 seconds of quad.
- The menu code came from id's CTF source code and its major problem was a tendency to cause client overflows. The menu code has been fixed to virtually eliminate overflows that were caused when iterating through the menu items.

For a quick test, try the following bindings if you have an Intellimouse (with the wheel):

```
> bind mwheelup invprev  
> bind mwheeldown invnext
```

Now on an L-Fire 1.x server or a Zoid CTF server, use your mouse wheel to scroll quickly through the Join menu. You'll overflow almost instantly. Now try the same test on an L-Fire 2.0 server and you should never overflow, no matter how fast you spin your mouse wheel.

Thanks to David "crt" Wright for tips on how to fix the id CTF menu overflow bugs.

- Clients could not open their Quake2 menu (hitting ESCAPE) when chasing another player. This was an id 3.20 source code fix.

5. Special Features

5.1. Improved Match Mode

The match mode in L-Fire 2.0 has been significantly improved over that of L-Fire 1.x. Specifically, L-Fire 2.0 aims to eliminate several annoyances that have occurred in the past when playing matches.

5.1.1. Independent Public Server and Match Mode Settings

One problem with a server that's shared between public server games and clan matches is that several of the settings that are appropriate for one mode are not appropriate for the other. For example, public servers usually have a `capturelimit` set to some value such as 8 whereas almost all matches take place with no `capturelimit` (`capturelimit` set to 0). Admins have to change the `capturelimit` every time the server goes in to or out of match mode. To solve this problem, match settings and public server settings in L-Fire 2.0 are totally independent from each other. The server has two sets of cvars – one set for match mode and one set for public server mode. Changing the settings for a match has no effect on the public server settings, and vice versa.

In addition to solving the problem of different settings between public and match mode, independent settings also allows server operators to allow match admins to change match settings, but they don't necessarily have to be given permission to change public server settings. For example, in L-Fire 1.x giving a player match admin permission resulted in that admin being able to change server settings at any time, even if the server was being used as a public server. In L-Fire 2.0, match admins can only change match settings – there is no longer a side-effect allowing them to also change settings for public server games.

To support independent settings, new commands have been added: `psettings`, `msettings`, `pset`, and `mset`. The `psettings` command shows the server's current public settings. The `msettings` command shows the server's current match settings. The `pset` and `mset` commands are used for changing public server and match settings, respectively. For example, "`pset capturelimit 5`" will change the public server's `capturelimit` to 5, while "`mset capturelimit 0`" will change the match mode `capturelimit` to 0.

5.1.2. Match Settings Displayed on the HUD

Match settings (`dmflags`, `weaponflags`, `timelimit`, etc) are displayed in the upper left corner of everyone's HUD during matchsetup. Any setting that is non-standard (such as the `mercylimit` being set to "2") is highlighted in green with the generally accepted standard setting shown to the right in parenthesis. This makes it easy for all of the players to verify that the correct match settings are in place before the match starts.

5.1.3. Locked Match Settings

Match settings are locked once the countdown has been started. This prevents anyone from changing the settings silently without anyone else knowing. There is no way to change the settings while a match is in progress, not even by using `rcon`.

5.1.4. Fixed Match Settings

Many settings that can be changed in public server mode are permanently set to some default value in match mode. A good example are the `sv_cheats` and `grapplesky` variables. These two server settings

still affect public server games as expected, but since it's unacceptable to activate/change them in any match play, they're disabled in match mode and there is no way to activate them. What this means to the players is that they'll always be guaranteed that no one has made subtle changes to the server settings which would affect match play.

5.1.5. MatchDefault Command

A new "matchdefault" command has been added that changes all of the match settings to reasonable match values at the same time. This is a convenience command for match admins to enable the admin to easily "fix" match settings that have been set to odd values.

5.1.6. Automatic Match Demos

Once the countdown starts, the server will automatically start demos for all of the players and spectators who have turned on the autodemo feature using the "autodemo" command.

5.1.7. Restricted Match Admins

L-Fire 2.0 supports restricted match admin permissions. Restricted match admins can only start and stop matches – they cannot change any match settings. This is useful if you run your match server for a specific tournament or league and don't want match admins changing the server's match settings.

You can give clients restricted match admin through the correct permission flag in the [admin] section of the lfire.ini file, or you can specify that elected match admins are restricted by setting the matchvote_restrictedadmin cvar to 1.

5.2. New Admin Permissions

The admin permissions have been changed in L-Fire 2.0. The admin permissions in L-Fire 1.x were too specific i.e. having separate admin flags for capturelimit and timelimit seems unnecessary. In L-Fire 2.0 the admin permissions have been redone so each permission flag represents a class of permissions instead of representing a specific command.

Here is a list of the new admin permissions:

Flag	Class	Description
B	Boot	Allows use of the boot command
C	Boot w/ Temp Banning	Allows use of the boot, bootban, and bootlevelban commands
D	Boot w/ Full Banning	Allows use of the boot, bootban, bootlevelban, and bootfullban commands
G	Gaming Variables	Allows changing the public server values for dmflags, weaponflags, fraglimit, capturelimit, mercylimit, timelimit, teambalance, ot_style, ot_sdtimelimit, ot_rarounds, hudtime, offhandhook, itemflags, and spawnprotect
I	Ignore Ping Restrictions	Allows the player to play on the server even if the player violates minimum or maximum ping policy
L	Level Control	Allows use of the changemap and nextmap commands

M	Match	Allows full match admin – matchsetup, matchstart, matchend, matchstate, lockteams, lockchat, mset, forceteams, changemap, and nextmap commands
P	Server Policies	Allows changing the antispam, minping, and maxping cvars
R	Restricted Match	Same as flag M (Match), except the mset command is not allowed so match settings cannot be changed
S	Chat Control	Allows use of the shutup and monitor commands
T	Team Control (CTF)	Allows use of the forceteam command
X	Priority Connect	Allows changing the priority_password, superuser_password, connectlimit_public, and connectlimit_priority cvars
Y	Server Password	Allows changing the password cvar
*	All Permissions	Assigns all of the individual permission classes to the admin

L-Fire 2.0 still supports the older permission flags, but you must keep your admin permissions in the LFireCfg/admin.cfg file, or you must put them in the [admin:legacy] section of your lfire.ini file. Note that if you have both an [admin] section in your lfire.ini file and an [admin:legacy] section, the server will only look for permissions in the [admin] section.

5.3. Intermission Map Voting

The intermission map vote is a menu that appears at the end of the level, letting your players decide for themselves what map they want to play next. The menu has three map choices that the players can choose from. The map with the most votes is the map that will be played next. In the event of a tie, the map that appears higher in the menu is the map that will be played.

There's two possibilities for determining which maps will appear in the menu:

- 1) If random map voting is active (maplist_random=1), then the server will choose three maps from your maplist at random. The server removes the five previous maps from consideration to prevent the same maps from being played too often. The server also eliminates maps that fail player restrictions (min/max player specifications on the maps).

You can adjust the probability that a map will appear in the menu by listing it more times in the maplist. Note that no matter how many times you list a map in the maplist, it still won't be a choice if the map was one of the past five maps played.

- 2) If random map voting is NOT active (maplist_random=0), then the server will pick choices sequentially from your maplist. For example, assume your maplist is q2dm1 through q2dm8 and the current map is q2dm2. When the voting menu appears, the choices will be q2dm3, q2dm4, and q2dm5.

By default, intermission map voting is off, set the intmapvote_allow cvar to "1" to turn this feature on.

It's recommended that you activate intermission map voting and random map selection. Together the two features have worked really well in testing on YYZ. Set intmapvote_allow to 1 and maplist_random to 1.

Important

If you are using the intermission map voting feature with random map selection, then it is highly recommended that your maplist have a minimum of about 10 maps. Please read this thread on the L-Fire message board for further explanation: <http://yyz.com/lfire/ubbx/Forum2/HTML/000038.html>

5.4. Priority Connect

Priority connect is a revolutionary feature that solves the problem of server admins and your server regulars not being able to connect to your server because too many newbies happen to be connected at the time.

To understand how priority connect works, understand that Quake2 has a single level of client limiting: the `maxclients` cvar. All clients are considered to have the same priority, and any client can connect as long as the server population has not yet reached `maxclients`. In other words, the server will let a player connect as long as the server population is less than `maxclients`. Once the server has a population of `maxclients`, the server rejects further connection attempts.

Instead of only having one connect level, the L-Fire 2.0 priority connect feature sets your server up with three connect levels: public, priority, and superuser. The client limiting on the three connect levels is defined by three cvars: `connectlimit_public`, `connectlimit_priority`, and `maxclients`. You also need two additional cvars to control access to the priority and superuser levels: `priority_password` and `superuser_password`.

The server uses the three connect levels as follows:

- 1) Any client can connect as long as there are less than `connectlimit_public` players on the server.
- 2) Once there are `connectlimit_public` players on the server, a client can only connect if he has a priority or superuser password. In the case of a client having a priority password and not a superuser password, the client can connect as long as there are less than `connectlimit_priority` players on the server.
- 3) Finally, once there are `connectlimit_priority` players on the server, a client can only connect if he has a superuser password.
- 4) The ultimate player limit is always defined by the `maxclients` cvar.

The easiest way to see how priority connect works in practice is by example:

```
set connectlimit_public 24
set connectlimit_priority 28
set maxclients 124

set priority_password "apple"
set superuser_password "orange"
```

Here we've set connect limits of 24/28/124. What this means is that any client can connect as long as there are less than 24 players on the server. In other words, the "public connect limit" is 24.

If there are 24 or more players on the server, a client can only connect if he has a priority or superuser password. Even with a priority password, the client can only connect if there is less than 28 players on the server. In other words, the "priority connect limit" is 28.

Finally, if there are 28 or more players on the server, a client can only connect if he has a superuser password. As stated earlier, `maxclients` defines the absolute player limit, so even a superuser cannot connect if there are 124 players on the server. Of course in practice this would never occur since you would have to have a ridiculous number of superuser clients connected to your server at the same time.

So how does a client set his priority or superuser password? By using the built-in "password" cvar. The player simply sets his password to match the server's `priority_password` or `superuser_password`. For example, if a client tries to connect to the above example server and there are 25 players connected, the client will be rejected and receive a message that says "Server is full (max 24)". If the client then sets his password to the text "apple" or "orange" and tries to connect, the server will allow the connection.

Pretty simple really. Players can just think of priority connect as "having a special password that allows connections to a full server."

We've been testing priority connect on YYZ-1 since November of '98 and it works very well. Your server regulars can be given priority passwords so they can connect when the server is close to full. Non-regulars will be rejected since they don't have the correct password. As a server admin with the superuser password, you will always be able to connect to your own busy server (assuming you set `maxclients` high enough).

What happens with priority connect in practice is that a busy server stays at or above the public connect limit. Let's assume in the above example that 25 players are connected to the server. At this population, only your regulars with a priority or superuser password can connect. As the non-regulars (who had connected when the server wasn't quite so full) disconnect from the server, only priority users will be able to connect and they will essentially take the place of the non-regulars. Eventually your server is populated mostly by the priority players and non-regulars that attempt to connect are rejected with a "server is full" type message. Priority players can still connect as long as the server population stays below `connectlimit_priority`.

If you decide to use priority connect, it's recommended that you set `maxclients` to 100 higher than your normal `maxclients`. This is simply for the sake of convention. As more servers start using priority connect the easiest way for people to tell if a server is full is by looking at the current player count and the `maxclients`. People will recognize that "124" really means "24" but the server is set up with priority connect. If the player sees 25/124 in GameSpy, he'll know that there are already 25 players connected and unless he has a priority password, he won't be able to connect because the effective public limit is 24 players. This little kludge is only necessary because GameSpy and the like don't support the concept of multiple connect levels. Until they do, the +100 convention is an easy way to communicate server availability to the players.

As a rough guide, you may want to set `connectlimit_public` to your previous `maxclients`, set `connectlimit_priority` to `connectlimit_public` + 4, and set `maxclients` to `connectlimit_public` + 100.

For example, with 16 `maxclients`, set your limits at 16/20/116. For 32 `maxclients`, set your limits at 32/36/132. You may want to tweak a bit by setting the public and priority limits one or two lower, such as 14/18/114 or 30/34/130. In practice, you'll find that `connectlimit_public` is almost never exceeded by more than a couple players, so you don't have to worry about setting your `connectlimit_priority` too high above `connectlimit_public`.

Finally, if YYZ is taken as a decent bellwether, you don't need to keep your server's priority password too much of a secret. Even when posted on a public message board, the priority password

doesn't get around too much. You'll find that most of your players won't use or won't know the priority password. You don't need to worry about everyone knowing the priority password and your server getting full of priority users. In many cases, at least half of your server's population is from players who aren't really "regulars" and they don't read your server's website or message board. A lot of players just scan servers through GameSpy and connect to the closest server that has some players. These are the sort of players that will be rejected when the server population exceeds `connectlimit_public`, leaving space available for your server's regulars.

5.5. L-Fire CTF and L-Fire DM integration

L-Fire CTF/DM is now released as a single binary (`gamex86.dll`, `gamei386.so`). There are no longer separate binaries for L-Fire CTF and L-Fire DM. The single mod runs in either CTF or DM mode depending on which directory the mod is installed in. If L-Fire CTF/DM is run out of the `quake2/ctf` directory, the mod automatically runs with CTF rules. If the mod is installed in any other directory (`baseq2`, `lfiredm`, etc.), the mod will run with DM rules.

This is all transparent to the players and server admins. The integration was done primarily to make mod compilation and distribution easier.

5.6. Map-Specific Config Files

L-Fire 2.0 allows you to create config files that will be executed for each map. If there is a config file in the mod directory named `<level>.cfg`, that config file will be executed when the level is loaded. The map-specific config file is executed no matter how the map is loaded, whether it's by a map command, `gamemap`, `changemap`, end of level map change, etc.

In addition to map-specific config files, a file in the mod directory named `"allmaps.cfg"` will be executed on every map load before the map-specific config file is executed. So you can put standard settings in `allmaps.cfg` and for odd maps where you may want to have a few custom settings, you can create custom config files just for those maps. Since the map-specific config file is executed after the `allmaps.cfg`, any settings you change in the map-specific config file will override settings in `allmaps.cfg`.

For example, let's say you normally want to ban the BFG from your server but for some odd reason you want to enable the BFG on `q2ctf4`.

In your `allmaps.cfg` you would put the following:

```
set weaponflags 512      // Ban BFG
```

In `q2ctf4.cfg` you would put this:

```
set weaponsflags 0      // No weapons banned
```

Since `allmaps.cfg` is executed before every map load, the `weaponflags` will be set to 512 for every map. When `q2ctf4` is loaded, `allmaps.cfg` will be executed (`weaponflags` set to 512), but then `q2ctf4.cfg` is executed where `weaponflags` will be set to 0. Now when `q2ctf4` starts, you'll see that the BFG is sitting there, waiting to spark the BFG llama-fest.

Finally, in order to implement this feature correctly, L-Fire 2.0 has to parse the config file manually (in other words, the mod does not rely on the Quake2 engine to execute the config file). So all this means to server admins is there's a few restrictions on what you can put in the config file. The following commands are valid in allmaps.cfg and the map-specific config files: echo, set, and exec. That essentially covers everything you would need to do for custom config files. The only issue that may come up is that you cannot set cvars by name only. In other words, the following statements are illegal:

```
hostname "My Server"  
password foo  
weaponflags 512
```

Instead, you must use the "set" command to change cvars:

```
set hostname "My Server"  
set password foo  
set weaponflags 512
```

A minor point, but something to be aware of.

Map-specific config files and maplist for WTF CTF (example)

Here's one more example of using map-specific config files and an advanced maplist to run regular CTF maps and WTF CTF maps at the same time. Normally you can't do anything like this because WTF CTF maps need to be run with unique settings: infinite ammo, very short timelimit (5 minutes), no capturelimit, etc. You also don't want to run WTF CTF maps when there's more than about 8-10 people on the server because the maps are too small for any more players.

So here's how to run standard CTF maps and interleave the WTF CTF maps as a short intermission between the standard maps.

Start by putting all of your standard variable settings in allmaps.cfg :

```
// allmaps.cfg  
set dmflags 280084  
set timelimit 30  
set capturelimit 8  
set mercylimit 5  
set fraglimit 0  
set supergrapple 0
```

Now set up a config file called wtfctf_all.cfg for WTF CTF maps:

```
// wtfctf_all.cfg  
set dmflags 288276 // 280084 plus infinite ammo  
set timelimit 5  
set capturelimit 0  
set mercylimit 0  
set supergrapple 1 // throw in Keyser grapple for fun
```

Now for each wtfctf map, execute wtfctf_all.cfg:

```
// wtfctf1a.cfg
exec wtfctf_all.cfg

// wtfctf2a.cfg
exec wtfctf_all.cfg

// wtfctf3a.cfg
exec wtfctf_all.cfg

etc.
```

Finally, set up a maplist in lfire.ini by listing the standard CTF maps interleaved with the WTF CTF maps:

```
[maplist]

=q2ctf1
=wtfctf1a, 0, 10
=q2ctf2
=wtfctf2a, 0, 10
=q2ctf3
=wtfctf3a, 0, 10
=q2ctf4a
=wtfctf4a, 0, 10
=q2ctf5
=wtfctf5a, 0, 10
```

Note that the maplist is defined so that the WTF CTF maps won't run if the player count is not between 0 and 10 players.

Another possible maplist:

```
[maplist]

=q2ctf1, 7, 0
=q2ctf2, 7, 0
=q2ctf3, 7, 0
=q2ctf4, 7, 0
=q2ctf5, 7, 0

=wtfctf1a, 0, 6
=wtfctf2a, 0, 6
=wtfctf3a, 0, 6
=wtfctf4a, 0, 6
=wtfctf5a, 0, 6
=wtfctf6a, 0, 6
```

This maplist runs the q2ctf maps if there are 7 or more players on the server, and it runs the WTF CTF maps if there are 6 or less players on the server. Note the "7, 0" after the q2ctf maps. The first number is the minimum number of players and the second number is the maximum. If the maximum is set to 0, then it means the maximum is ignored.

So you can see how you can now do things with your servers that have never been possible before, and are not possible with any other mod.

6.New CVars

6.1. New CVars for both L-Fire CTF and L-Fire DM

6.1.1. **matchvote_restrictedadmin** [0/1]

Specifies whether elected match admins have restricted permissions. Restricted permissions allow match admins to start and stop matches, but they are not allowed to change any match settings, such as dmflags, timelimit, etc. Defaults to 0 (elected match admins are not restricted).

6.1.2. **forcejoin** [0/1]

Specifies whether players are automatically inserted into the game when they connect to the server. In DM, players are automatically spawned into the game. In CTF, they are automatically placed on a team and spawned into a base. This feature is useful on newbie servers where players don't know how to advance past the message of the day and Join menus. Defaults to 0 (no force join).

6.1.3. **intmapvote_allow** [0/1]

Specifies whether intermission map voting is allowed.

0 = no intermission map voting (default)

1 = map vote at the end of each level

6.1.4. **intmapvote_time** [time_secs]

Specifies the time in seconds for the intermission map vote. Defaults to 20 seconds.

6.1.5. **execlevelconfigs** [0/1]

Controls whether map-specific config files will be executed at the start of each level.

0 = don't execute map specific config files

1 = execute map specific config files (default)

6.1.6. **forcemode** [0/1/2]

Forces the server to run with DM or CTF rules, regardless of what directory the mod is installed in. There should be no reason to use this cvar unless you are doing odd testing.

0 = mode automatically determined based on game directory (default)

1 = force DM mode

2 = force CTF mode

6.1.7. **logdirectory** [directory]

Specifies the directory for logfiles. Default is based on the mod directory with "_logs" appended (baseq2_logs, ctf_logs, lfredm_logs, etc.)

6.1.8. **connectlimit_public** [public_limit]

6.1.8. **connectlimit_priority** [priority_limit]

These cvars specify the public and priority connect limits. Both default to 0 (no priority connect in use).

6.1.9. **priority_password** [password]

6.1.9. **superuser_password** [password]

These cvars specify the priority and superuser passwords, respectively. Both default to nothing (no

priority connect in use).

6.1.10. needpass

id 3.20 serverinfo cvar indicating whether a password is needed to connect to the server. This cvar is automatically set by the server.

0 = no password needed (default)

1 = need correct password

6.1.11. filterban [0/1]

Controls whether addresses that match in the IP filter lists result in client acceptance or refusal.

0 = reject clients unless they match an IP filter

1 = reject clients if they match an IP filter i.e. IP ban (default)

6.1.12. flood_msgs [count]

id 3.20 antiflood cvar. Maximum number of messages allowed in the time period specified by flood_persecond. Default is 0 (id flood control off).

6.1.13. flood_persecond [time_secs]

id 3.20 antiflood cvar. Time period that a maximum of flood_msgs messages are permitted. Defaults to 4 seconds.

6.1.14. flood_waitdelay [time_secs]

id 3.20 antiflood cvars. Amount of time a client gets muzzled for chat flooding. Defaults to 10 seconds.

6.1.15. sv_maplist [list of maps]

id 3.20 cvar. Specifies the server's maplist. Default is blank, meaning no maplist.

Note that the server first checks sv_maplist and if it doesn't find one, it checks for a maplist in the lfire.ini config file. It's recommended that you use the lfire.ini maplist instead of sv_maplist since the lfire.ini maplist is much more sophisticated.

6.1.16. maplist [0/1]

Specifies whether the server will use the maplist in the lfire.ini file (or the legacy maploop.cfg if there's no lfire.ini file). Remember, as stated above, sv_maplist takes precedence over the L-Fire maplist if both are defined. Defaults to 1.

Note that the old "maploop" cvar still works and does the same thing as the maplist cvar. If either maploop or maplist is active, then the server will potentially look in the lfire.ini file (or the legacy maploop.cfg).

6.1.17. maplist_random [0/1]

Specifies whether maps will be chosen at random from the lfire.ini maplist. Defaults to 0.

6.1.18. mapvote_maxclients [num_clients]

Specifies the maximum number of clients to allow a console map vote (mapvote command). Defaults to 0 (no maximum).

6.1.19. mapvote_minclients [num_clients]

Specifies the minimum number of clients required on the server before a mapvote can be started.

Defaults to 1.

6.1.20. mapvote_pass [percentage]

Minimum percentage of YES votes required for the mapvote to pass. Note that any clients who do not vote are counted as a NO vote. Defaults to 51.

6.1.21. mapvote_allow [0/1]

Specifies whether map voting is allowed. Defaults to 0 (off).

6.1.22. mapvote_time [time_secs]

Specifies the time in seconds that the map voting period is active. Defaults to 30 seconds.

6.1.23. mapvote_clientlimit [count]

Specifies the maximum number of times per level that a client may initiate a map vote. Defaults to 3.

6.1.24. showplayercounts [0/1]

Specifies whether player counts are displayed whenever players connect, disconnect, change to spectators, etc. Defaults to 1.

6.1.25. forward_address [address]

Used for automatic player forwarding to another server. Specifies the address that players will be forwarded to, such as "192.168.0.1:27910". Defaults to nothing (no forwarding).

6.1.26. forward_servermessage [message]

Used for automatic player forwarding to another server. Specifies custom text for the forwarding address in the message that the player gets when he is being redirected to the new server. Defaults to nothing, in which case the message will contain the same address as specified in the forward_address cvar.

6.2. New CVars for L-Fire CTF Only

6.2.1. arenaintermission [0/1/2/3]

Specifies whether a Rocket Arena or RailWar intermission will be played after each level.

0 = no intermission (default)

1 = Rocket Arena intermission

2 = RailWar intermission

3 = Pick intermission randomly

6.2.2. arenaintermission_mercylimit [count]

Specifies a mercylimit for determining whether intermission will be played. If the difference in captures equals or exceeds this value, then intermission won't be played at the end of the level. Defaults to 0, which means captures are irrelevant for determining whether intermission will be played.

6.2.3. showpingrating [0/1]

Specifies whether team ping rating is displayed on the scoreboard. Defaults to 1.

6.2.4. resetuneventeams [0/1]

Specifies whether teams are reset (all players moved to spectator) if the previous level ended in a blowout. Defaults to 0.

6.2.5. levelstart_reset [time_secs]

Specifies the time in seconds for the level start countdown if the teams had been reset due to blowout. Defaults to 60 seconds.

6.2.6. freeteamchange [0/1]

Specifies whether players can change teams for free. Defaults to 0.

6.2.7. freeteamchange_protect [time_secs]

When freeteamchange is in effect, specifies the time in seconds that a player is invulnerable when he changes teams. This is a good idea since a player may change teams to help out team balancing and find himself in the middle of a lot of enemies (after all, he *is* changing to the losing team which may have their base taken over). Defaults to 5 seconds.

6.3. New CVars for L-Fire DM Only

6.3.1. gameversion [version_info]

Serverinfo cvar that contains the mod name and version, such as "L-Fire DM v2.0". This information used to be contained in the gamename cvar, but that caused the CLQ and some GameSpy filters to filter out L-Fire DM servers. This should solve the problem. The server sets this cvar for you automatically.

6.3.2. saveffscores [0/1]

Specifies whether a fastest frags list is saved for each level.

Note that turning on fastest frags automatically disables the highscores display (no point in having both). Defaults to 0.

7. New Commands

7.1. New Client Commands

7.1.1. matchdefault

Used in match mode, this command sets all of the match mode server variables to reasonable default values.

7.1.2. playerlist

id 3.20 command. List the players, how long they've been playing, whether they are spectating, and who they may be chasing.

7.1.3. mapvote <level>

Starts a vote to change the map. If mapvote is used without any arguments, then the server will give the player a list of valid maps to vote for.

The map is validated against the lfire.ini [changemap] section.

7.1.4. autodemo

Toggles autodemo for the player. Autodemo automatically starts and stops demos for the player as the server changes levels.

7.1.5. chaseme

Tells the player which spectators are currently chasing him.

7.1.6. chathud

Toggles the team message chat display.

7.1.7. psettings

Lists the public server settings (dmflags, timelimit, capturelimit, etc.)

7.1.8. msettings

Lists the match mode settings (dmflags, timelimit, capturelimit, etc.)

7.1.9. pset <variable> <value> (Admin only)

Sets a public server variable. Giving pset a variable name without a value to set will display the current value for the variable.

7.1.10. mset <variable> <value> (Admin only)

Sets a match mode variable. Giving mset a variable name without a value to set will display the current value for the variable.

7.1.11. monitor (Admin only)

Toggles chat monitor mode.

7.1.12. ffscores <level> (DM only)

Displays the fastest frag scores for the specified level. Analogous to the "highscores" command.

7.2. New Server Commands

7.2.1. sv addip**7.2.1. sv removeip****7.2.1. sv listip****7.2.1. sv writeip**

id 3.20 commands for IP filtering.

7.2.2. sv loadipfilter

Reloads the ipfilter section in the lfire.ini configuration file. Note that the old "sv loadban" command still works and does the same thing as "sv loadipfilter".

7.2.3. sv loadmaplist

Reloads the maplist section in the lfire.ini configuration file. Note that the old "sv loadmaploop" command still works and does the same thing as "sv loadmaplist".

7.2.4. sv nextmap

Runs the next map in the maplist, or picks a new random map if maplist_random is active. Useful for testing random maplists.

7.2.5. sv changemap

Same as the client "changepmap" command.

8. Upgrading From Previous L-Fire 2.0 Betas

This section is intended for server admins who are currently running an L-Fire 2.0 beta server and want to upgrade to L-Fire 2.0 beta 125. You do not need to read this section if you are not currently running an L-Fire 2.0 beta server.

8.1. Important Upgrade Instructions (2.0b110 and 2.0b115)

8.1.1. Configuration File Changes (LFire.ini)

If you have upgraded your server to use the new lfire.ini config file scheme, then the first thing you must do before running this latest beta is to change your lfire.ini [admin] section to [admin:legacy].

For example, if your [admin] section looks like this:

```
[admin]
=joe/          BCDE
=hdtv/        *
=matchadmin/   P
```

Change your admin section to look like this:

```
[admin:legacy]
=joe/          BCDE
=hdtv/        *
=matchadmin/   P
```

The section name must be changed to [admin:legacy] because the admin permission flags in this latest beta have totally changed from previous betas and all previous versions of L-Fire CTF. The server will now consider permissions in the [admin:legacy] section to be the old values for the permission flags, and the [admin] section will have the new values for permission flags.

See the instructions in section 5.2 (New Admin Permissions) on how to migrate your old admin permissions from the [admin:legacy] section to the [admin] section.

8.1.2. Intermission Map Voting

An important note about map voting.

If you are using the intermission map voting feature with random map selection, then it is highly recommended that your maplist have a minimum of about 10 maps. Please read this thread on the L-Fire message board for further explanation: <http://yyz.com/lfire/ubbx/Forum2/HTML/000038.html>

8.2. Changes and New Features for Both L-Fire CTF and L-Fire DM

This section summarizes changes and new features that have been added since L-Fire 2.0 beta 115.

- Added in-eyes chase camera. Thanks to David “crt” Wright for the code.

The chase/eye camera can be selected from the Join menu and then you can toggle between the chase/eye camera modes when you’re a spectator by pressing the FIRE button.

- Match mode has been significantly enhanced. See details in section 5.1 of this document

(Improved Match Mode).

- Added player forwarding. This feature is used when you're moving your server to a new IP address or port and you want the server at the previous address to automatically forward all connecting clients to the new server.

To use this feature, set the “forward_address” cvar to the new address, such as “192.168.0.1:27910”, and then restart the server. When players connect to your server they will see the message of the day and will be automatically reconnected to the new address when they try to advance past the message of the day (by pressing ENTER, TAB, ESCAPE, etc.)

You should put some text in your message of the day like this: “SuperAwesome DM has moved to 192.168.0.1 (quake.superawesome.com). Please update your address books with the new server address.”

When the server stuffs a connect command to the player, the player will also get the following message: “You are being forwarded to <address>”. <address> is the address specified in the forward_address cvar. You can replace <address> with your own text such as “192.168.0.1 (quake.superawesome.com)”. Then the player will get a message like this: “You are being forwarded to 192.168.0.1 (quake.superawesome.com)”.

- Added forcejoin cvar. Setting forcejoin to 1 will cause players who connect to the server to immediately join the game. In DM, players will be spawned into the level. In CTF, players will be given a team and spawned into their base.
- Added support for info_position entities (mostly useful for CTF or Team DM). These are entities that contain location information for the map so instead of the %L chat macro expanding to a string like “by the red shotgun”, it will expand to a string like “near the red railroom”.
- Added player shutup by IP address. This is supported in the [ipshutup] section of the lfire.ini file. The IP address format is the same as in the [ipfilter] section. Note that the “sv loadshutup” command will now reload both the [shutup] and [ipshutup] sections of the lfire.ini file.
- Added support for adding additional entities to a map instead of having to do a full replacement of the bsp’s entities. In L-Fire 1.x, a <level>.ent file would replace all of the bsp’s entities with the entities specified in the ent file. Now you can put #add at the beginning of your ent file and this will instruct the server to add the file’s ents to the ones in the bsp, and not do a full replacement. So if all you wanted to do is add a power shield to a map, your ent file would look something like this:

```
#add
{
  "origin" "700 200 -400"
  "classname" "item_power_shield"
}
```

- Changed the id gender detection code so players with custom skins (non-male/female) will no longer be referred to as “it”. For example, the server will now say “Joe blew himself” up instead of “Joe blew itself up” if the player Joe has a non-male/female skin.
- You can now clear the priority_password and superuser_password by setting them to “none”. This is also how you can clear the password cvar.

- Newlines are filtered from chat text so mischievous players can't do "anonymous" chatting.
- Added killer's average ping statistic to stdlog frag output.

8.3. Changes and New Features for L-Fire CTF Only

This section summarizes CTF changes and new features that have been added since L-Fire 2.0 beta 115.

- Scoreboard now indicates "Match Mode" if the server is in match mode.
- Added team chat HUD. This is an experimental feature to help CTF teamplay. One of the problems with say_team chat text is that many times your teammates' chat text scrolls off the console before you've had a chance to read it. This is especially true if the chat text comes when you're busy fighting or a lot is happening and obituary messages are coming quickly. The chat HUD aims to solve this problem by saving a 30 second history of chat messages. The messages are displayed in the lower left corner of the players' screens. For players with at least 800x600 screen resolution, the chat text doesn't intrude much on the HUD. For people playing with a lower resolution, a key can be bound to toggle the chat HUD only when the player wants to see the team chat buffer.

The advantage of the chat HUD over bringing down the console to see old chat text is obvious. Instead of being vulnerable as half your screen is covered by the console and not being able to move either, you can move around and read the chat text at the same time. The disadvantage of the chat HUD is that you really need a minimum of 800x600 resolution so it doesn't intrude too much on the player's screen. At 1024x768 resolution or above, the chat HUD is almost a no-brainer as it hardly intrudes on the player's screen at all.

Toggle the chat HUD by using the "chathud" command. By default, the chathud is 50 columns by 8 rows and the history is 30 seconds. At this time the chat HUD is not player configurable, but it will be in a future release. For example, players running at 1024x768 or above might want a 80x10 display with a 60 second history.

Depending on player feedback during beta testing, this feature may or may not be included in the final L-Fire 2.0 release.

- Removed team balancing. Sorry – it was too controversial and whether you liked it or hated it, too many players were bitching over it.
- Forcing a player to spectator no longer triggers the team change flooding detection. This was a problem for admins as they could only force players to a team/spectator a limited number of times before the team change flood detection kicked in and prevented further forceteam commands on the player.
- Player scores are now reset to zero at the beginning of each rocket arena overtime round, making it much easier to tell who did well during the round.
- Removed the colorquad cvar. This option is no longer necessary as colored quad (red/blue glow depending on team) is now standard across all CTF mods.
- Added %X chat macro which expands to the location of all techs that are in the player's line of

sight. For example, the text “Tech spotted: %X” may expand to “Tech spotted: Autodoc near the red flagroom, Power Amp in the red main entrance”. Note that the location is that of the tech, not the player sending the chat message.

- Added %C chat macro which expands to the location of any flag carriers that are in the player's line of sight. For example, the tech “%C” may expand to “Friendly flag carrier spotted in the water near the red water tube”. Note that the location is that of the flag carrier, not the player sending the chat message.
- Added %F chat macro which expands to "a Red Flag" or "a Blue Flag" as appropriate if the chatting player is currently carrying the red/blue flag.
- Added %Q chat macro which expands to "Quad" if the chatting player currently has quad.

8.4. Bug Fixes

This section summarizes bug fixes that have been made since L-Fire 2.0 beta 115.

- If a map does not have an entry in the mapnames.ini file or the mapnames.ini files does not exist, then the map name such as "q2ctf1", "q2dm1", etc. is now used in the intermission voting menu instead of being blank.
- Fixed bug where teams would not reset after a blowout if rocket arena intermission or railwar intermission was used.
- Fixed bug where too many maps in the [changemap] section of the lfire.ini file could cause the server to crash if a client used the changemap or mapvote commands without any arguments. Thanks to Sentry for reporting the bug.
- Fixed an id CTF bug where the %L chat macro would indicate "in the water" if the player was in water, slime, or lava. The fixed code now correctly says "in the water", "in the slime", or "in the lava" as appropriate.
- Fixed an id bug where weapons in a loop don't rotate if weapons stay is on. For example, in Tokay's Towers there is a railgun and BFG that are supposed to randomly rotate when the weapon is picked up. When weapons stay is on, only the railgun appears. In L-Fire 2.0, the weapon does not stay when picked up, and a random weapon is chosen when the weapon respawns.

[Today's Active Topics \(all public forums\)](#)

You last visited: 07-24-99 11:06 AM

Registered Members: 202

	Forum	Posts	Last Post	Moderator
Quake Mods				
	L-Fire CTF Forum for discussions, suggestions, and bug reports regarding L-Fire CTF.	524	07-24-99 11:11 AM	
	L-Fire DM Forum for discussions, suggestions, and bug reports regarding L-Fire DM.	106	07-22-99 05:57 PM	
	Weapons of Destruction Forum for discussions, suggestions, and bug reports regarding Weapons of Destruction.	83	07-21-99 05:02 PM	
Misc				
	Level Editing Forum for discussions about level editing and Hash Assassin's tutorials.	2	01-22-99 10:50 AM	

All times are PT (US). All dates are in MM-DD-YY format.

- [New Posts Since Your Last Visit](#)
- [No New Posts Since Your Last Visit](#)

[Contact Us](#) | [L-Fire Development](#)Powered by: Ultimate Bulletin Board, Version 5.26
© Madrona Park, Inc., 1998-1999.