BF3 PC Server Administration

This is the server admin manual for BF3 PC Server ~~R20~~R21.

Contents

Game server operation

When the game server first starts up, it reads a set of configuration files from disk. Some of these are managed by the RSP, and some by the server administrator.

The game server will then cycle through a series of maps. Game clients can connect to the server and play on the maps.

Control of the game server is done through a "Remote Administration" interface. This is a TCP port (kind of like a terminal interface). There are both Python scripts and GUI tools which control the game server through this mechanism.

Players can indirectly communicate with the GUI tools by sending special chat commands, which the GUI tools react upon.

The game server writes a set of log files to disk while it is running; these can be inspected by the server admin.

Files which may be accessible to the server admin

EA decides which files the RSP may make available to the admin. It is up to each RSP how to facilitate this, and the extent to which access is given. The list below contains all the files which the RSP is allowed to give the server admin full access to:

Admin/*.txt
pb/svss/*
pb/svlogs/*
pb/*.cfg
pb/pbbans.dat
pb/pbucon.use
pb/sv_viol.log
pb/sv_cheat.log

In addition, the server admin may have limited access to a few lines in ServerOptions.cfg and the values of some commandline arguments.

Remote administration interface

The remote administration interface is a two-way channel for sending and receiving commands from the game server. Before the remote administration can be used, a remote admin password must be set, either via Admin/Startup.txt, ServerOptions.cfg or the commandline.

Do notice that the remote admin interface is normally case sensitive.

Commands

There's a Python script, called "CommandConsole.py", that can be used to connect to the remote administration interface. Once connected, there is an assortment of commands available that can be sent. See "BF3 PC Remote Administration Protocol.pdf" for the full list.

Events

The game server can also send events when specific things happen in-game. For instance, when a player joins or leaves the server, when a round ends, or when anyone says anything through the chat. The Python script called "EventConsole.py" can be used to listen to these events.

GUI tools

There are several GUI tools constructed, which make it easier to control the game server. We'd recommend that you use them rather than relying on CommandConsole.py / EventConsole.py for everyday use.

Startup script

The game server will process the file named Admin/Startup.txt during bootup. Each line in that file will be executed as a remote administration command.

Password

Anyone who knows the IP address and port of the remote administration interface of a server can connect to it and retrieve some basic information, including a player list. Most commands require the user to specify a password. This password can be changed by issuing the **admin.password** command.

Usually the **admin.password** command is put into the server's static configuration.

When do commands take effect?

Some commands take effect immediately (example: kicking a player).

Some commands take effect only after a round change.

Some commands must be put into Admin/Startup.txt to take effect at all.

Accounts, soldier names, and GUIDs

Every BF3 PC player has exactly one EA account. The player has exactly one soldier name.

The PunkBuster GUID is tied to the EA account. So is the "EA GUID".

The PB GUID is used with all PB-services, while the EA GUID is used with any non-PB-related functions in the game server.

PunkBuster GUIDs are 32-digit hexstrings.

EA GUIDs are the prefix "EA_" followed by a 32-digit hexstring.

Player slots

How many players does a server support?

This is determined by two factors:

- The RSP has a max-cap which they can set per server
- The admin also has a max cap that can be set (**vars.maxPlayers**)

The current max number of players is the minimum of the both.

Ban system

The game server has an internal ban system. This system is independent from PunkBuster's banlist. At startup, ban entries are read from the file named banlist.txt. During runtime, the **banList.*** commands can be used to manipulate the banlist.

Players can be banned either on their soldier name, or on their EA GUID. Banning someone on their soldier name is not particularly effective – if it's a determined griefer then he/she will just create a new soldier and return . Banning someone on their EA GUID is much more effective.

To find out someone's EA GUID, perform admin.serverInfo while that person is playing on your server. Or – inspect the AdminLog.

People can be banned either for a few seconds, until the end of the current round, or permanently.

The banlist can contain up to 10.000 entries.

banlist.txt format

Each entry in the banlist occupies 5 lines.

The first line specifies what the ban is on:

guid – ban on EA GUID
name – ban on soldier name
ip – ban on game client IP address

The second line specifies the GUID/name/IP that the ban applies to

The third line specifies the duration of the ban:

perm – permanent
round – until the end of the current round
seconds – until the given time is reached

The fourth line contains the timestamp for a "seconds"-type ban; otherwise it is unused.

The fifth line contains the reason for being banned. Max length 80 characters.

Map handling

Overview

BF3 PC game servers are designed to rotate through a sequence of maps. The exact configuration is specified in the server's internal map list. Different maps in the map list can use different game modes. (Note however, that some game modes will not work properly if there are more players on the server than the game mode is designed for.)

Upon startup, the Admin/MapList.txt file is read. During runtime, the **mapList.*** commands can be used to edit the set of maps.

When the same map is played for several rounds, all 2-team gamemodes stipulate that the teams will switch sides after a run. This way, a 2-round session of Rush will have players play both attackers and defenders.

Controlling map switching

mapList.* can be used to edit the maplist while the server is running.

mapList.setNextMapIndex sets which will be the next map.

mapList.getMapIndices returns information on which is the current and next map in the list.

mapList.runNextRound switches to the next round, without finishing the current.

mapList.restartRound makes all players reload the current map, and restarts the current round.

mapList.endRound declares a specific team as the winning team, and then moves directly to the end-of-round screen.

Admin/MapList.txt format

Each line in the file has three entries: the map name, the game mode, and the number of rounds to be played on the map until proceeding to the next map in the list.

Example MapList.txt:

```
MP_007 RushLarge0 2
MP_018 ConquestSmall0 2
MP_Subway RushLarge0 4
```

Idle timeout

If a player doesn't give any input within a specific period of time, he/she will be kicked due to idling. You can change the time interval / disable the idle timeout through **vars.idleTimeout**. In addition, **vars.idleBanRounds** can be used to apply a ban for a number of rounds for someone that gets kicked due to idle timeout.

Warm-up and pre-rounad

When a server first starts up, there are no players on it and the server is in warm-up state. Warm-up is a state where players can move around, complete objectives and so on, but scoring is disabled. Once the required number of players is reached, the game will reset the level and transition into pre-round. In pre-round, a timer counts down to the round start; players cannot move, shoot or take objectives during pre-round. When the timer has run out, the actual round begins. Players can move freely, take objectives and scoring is enabled during the round. If the number of players drops beneath the minimum threshold during the round, the round will be aborted and the server switches back to warm-up.

You can change the number of players requires to go between warm-up and in-round using **vars.roundStartPlayerCount** and **vars.roundRestartPlayerCount**. The start player count must be higher than the restart player count – so if you set the starting player count below the restart player count, the engine will silently assume that the restart player count is one lower than the start player count.

Ranked and unranked servers

When a server starts up, it will default to being ranked. You can use **vars.ranked false** to make it unranked during startup. Once the server has been switched to unranked, it cannot be switched back to ranked. The **vars.ranked** command is best placed at the top of Admin/Startup.txt.

Ranked servers will automatically run with PunkBuster enabled.

Below is a list of all settings and when they can be changed in relation to server being ranked/ unranked.
"Always Allowed": The value can be changed at all times for both Ranked and Unranked servers.
"ReadOnly After Startup": Can only be changed during server startup.
"ReadOnly On Ranked": Can only be set/changed on Unranked servers.
"Disallowed On Ranked": Can only be invoked on Unranked servers.

Setting	When this can be changed
login.plainText	Always Allowed
login.hashed	Always Allowed
logout	Always Allowed
quit	Always Allowed
admin.help	Always Allowed
admin.shutDown	Always Allowed
admin.password	Always Allowed
admin.eventsEnabled	Always Allowed
vars.serverName	Always Allowed
vars.gamePassword	Disallowed on Ranked, ReadOnly After Startup
vars.ranked	ReadOnly After Startup [*1]
version	Always Allowed

serverInfo	Always Allowed
listPlayers	Always Allowed
currentLevel	Always Allowed
admin.listPlayers	Always Allowed
admin.movePlayer	Always Allowed
admin.kickPlayer	Always Allowed
admin.say	Always Allowed
admin.killPlayer	Always Allowed
banList.add	Always Allowed
banList.remove	Always Allowed
banList.clear	Always Allowed
banList.list	Always Allowed
banList.load	Always Allowed
banList.save	Always Allowed
gameAdmin.add	Always Allowed
gameAdmin.remove	Always Allowed
gameAdmin.clear	Always Allowed
gameAdmin.list	Always Allowed
gameAdmin.load	Always Allowed
gameAdmin.save	Always Allowed
mapList.add	Always Allowed
mapList.remove	Always Allowed
mapList.clear	Always Allowed
mapList.list	Always Allowed
mapList.load	Always Allowed
mapList.save	Always Allowed
mapList.setNextMapIndex	Always Allowed
mapList.getMapIndices	Always Allowed
mapList.getRounds	Always Allowed
mapList.runNextRound	Always Allowed
mapList.restartRound	Always Allowed
mapList.endRound	Always Allowed
mapList.availableMaps	Always Allowed
reservedSlotsList.add	Always Allowed
reservedSlotsList.remove	Always Allowed
reservedSlotsList.clear	Always Allowed
reservedSlotsList.list	Always Allowed
reservedSlotsList.load	Always Allowed
reservedSlotsList.save	Always Allowed
unlockList.add	Always Allowed
unlockList.remove	Disallowed On Ranked
unlockList.clear	Disallowed On Ranked
unlockList.list	Disallowed On Ranked
unlockList.save	Disallowed On Ranked
vars.teamKillCountForKick	Disallowed On Ranked
vars.teamKillValueForKick	Always Allowed
vars.teamKillValueIncrease	Always Allowed
vars.teamKillValueDecreasePerSecond	Always Allowed
vars.teamKillKickForBan	Always Allowed
vars.idleTimeout	Always Allowed
vars.idleBanRounds	Always Allowed

vars.autoBalance	Always Allowed
vars.roundStartPlayerCount	Always Allowed
vars.roundRestartPlayerCount	Always Allowed
vars.serverMessage	Always Allowed
vars.serverDescription	Always Allowed
~~**vars.killRotation**~~	~~Always Allowed~~
vars.killCam	Always Allowed
vars.roundsPerMap	Always Allowed
vars.bannerUrl	Always Allowed
vars.vehicleSpawnAllowed	Always Allowed
vars.vehicleSpawnDelay	Always Allowed
vars.soldierHealth	Always Allowed
vars.playerRespawnTime	Always Allowed[*4]
vars.3pCam	Always Allowed
vars.regenerateHealth	Always Allowed
vars.friendlyFire	Always Allowed
vars.playerManDownTime	Always Allowed
vars.bulletDamage	ReadOnly On Ranked
vars.minimap	ReadOnly On Ranked
vars.hud	Always Allowed
vars.miniMapSpotting	Always Allowed
vars.3dSpotting	ReadOnly On Ranked
vars.nameTag	Always Allowed
vars.gameModeCounter	Always Allowed
vars.onlySquadLeaderSpawn	Always Allowed
vars.maxPlayers	Always Allowed
vars.unlockMode	Always Allowed
punkBuster.activate	ReadOnly On Ranked
punkBuster.isActive	Always Allowed
punkBuster.pb_sv_command	Always Allowed
	~~Always Allowed~~

[*1] Can only go from true to false

[*4] Clamped to 60-100 on ranked

Reconfiguring the game modes

This is done through a large number of settings.

Changing these settings will change how your server gets listed in the browser; see [battlelog info] for more information.

The following settings are available:

vars.serverName controls the name of the server, as seen in the server browser.

vars.gamePassword - if set, players must enter this password when connecting to the server.

vars.friendlyFire – when set, people can inflict damage on others in the same team.

vars.killCam – when set, a killed player gets to see a close-up of his/hers killer for a few seconds.

vars.miniMap – when set, a minimap is available in the bottom-left corner of the screen during play.

vars.hud – when set, the hud is present.

vars.crossHair – when set, guns have crosshairs in the center of the screen.

vars.3dSpotting – when set, spotted targets are marked with icons in the 3D world.

vars.miniMapSpotting - when set, spotted targets are marked with icons on the minimap.

vars.3pCam - when set, 3rd person vehicle cameras are enabled.

vars.nameTag – when set, nametags are rendered over players' heads in the 3D world.

vars.regenerateHealth – when set, health regeneration is enabled.

vars.vehicleSpawnAllowed – when set, vehicles will spawn in-game.

vars.vehicleSpawnDelay – controls the delay between vehicle spawn; specified in percent (100% = normal).

vars.soldierHealth – sets maximum soldier health, specified in percent (0-100%; 100% = normal).

vars.playerRespawnTime – controls player respawn delay; specified in percent (100% = normal).

vars.playerManDownTime - controls player man-down time; specified in percent (100% = normal).

vars.bulletDamage – controls bullet damage; specified in percent (0-100%; 100% = normal).

vars.onlySquadLeaderSpawn – when set, players can only spawn on the squad leader.

vars.unlockMode – controls which unlocks will be available to players on an unranked server.

vars.roundStartPlayerCount – when the server is in pre-round, it waits for this many players to be present until it proceeds to start the real round. This value must be higher than the roundRestartPlayerCount.

vars.roundRestartPlayerCount – when a round is going, if the number of players drops under this number, the round will be aborted and the server moves back to pre-round.

vars.gameModeCounter – controls the number of tickets required to end round (100% = normal).

Unlocks and stats / progression

Players' progression in MP are stored on the central servers. A ranked server will always fetch the stats for a player from the central server when the player joins; these stats control which weapons and gadgets are available to the player. Any new stats and new unlocks are written back to the central servers when the player disconnects from the server.

An unranked server offers several different options when it comes to which set of weapons is available. Common to these is that players will see score when playing on such a server, but their stats will never get written back to the central servers.

Which set of unlocks is available on an unranked server is controlled by **vars.unlockMode.** The following values are available:

Setting	Weapons/gadgets available
all	All weapons which the player could possibly unlock by playing ranked games; this includes weapons in any DLCs which the player has purchased
common	The set of weapons which is not part of any DLCs or special campaigns; this is a "safe" subset for matches
stats	The same set of weapons which the player has available in a ranked match
none	Only the base set of weapons

Server settings and Battlelog

Battlelog has a set of filters that are designed to help players find a couple of standard-configured game servers.

Quickmatch

First off, there is the Quickmatch operation.

Quickmatch will only go against ranked servers. Also, the ping site setting is taken into consideration; players will have a slight preference to game servers which belong to the player's closest ping site location.

If the user has not selected any filters in the server browser filter, then a couple of default search criteria will be applied:

- Preset: Normal (see below)
- Game Mode: Rush, Conquest, Team Deathmatch
- Game: Base game only (not Back to Karkand)

If you select any items in the following categories in the server browser filter, then they replace the corresponding default criteria:

- Preset
- Game Mode
- Maps
- Game (base / Back to Karkand)

Other categories have no effect on quickmatch.

Server browser

Then there is the server browser with its filters. One of the filter sections is the preset selection. By default, no filtering is done on presets (so the server browser show all servers regardless of any game mode-affecting settings).

There are four different presets in Battlelog: Normal, Hardcore, Infantry Only and Custom.

If your server has settings set to match either Normal, Hardcore or Infantry only, then it will show up in matching searches. If your settings are different, then it will match a search for Custom servers.

Setting	Value for Normal	Value for Hardcore	Value for Infantry Only
vars.autoBalance	true	true	true
vars.friendlyFire	false	true	false
vars.killCam	true	false	true
vars.miniMap	true	true	true
vars.hud	true	false	true
vars.3dSpotting	true	false	true
vars.miniMapSpotting	true	true	true
vars.nameTag	true	false	true
vars.3pCam	true	false	false
vars.regenerateHealth	true	false	true
vars.vehicleSpawnAllowed	true	true	false
vars.soldierHealth	100	60	100
vars.playerRespawnTime	100	100	100
vars.playerManDownTime	100	100	100
vars.bulletDamage	100	100	100
vars.onlySquadLeaderSpawn	false	true	false

Expansion packs

Servers can have levels from multiple expansions in their map rotation. The set of used expansion packs will be reported to Battlelog and exposed in the server browser.

Per default, all servers will be visible in the server browser, independent of map rotations or expansion packs. Only once players use the 'game' filter to look for specific expansions will servers be culled based on the expansion packs they use.

Note however, that this filtering is strict, so if the server requires additional expansion packs which the player hasn't selected, that server won't be seen in the server browser. For instance a server running original & Karkand maps would show up if a player doesn't filter on any expansion or selects both 'Battlefield3' and 'Expansion pack - Back to Karkand' in the server browser. Selecting only one of them would not return that server since it requires both.

For players joining a game server, the addition of expansion packs will present 2 special cases where a player is missing one or more expansion packs used by the server:

- If the current level is available to the client, a warning will be displayed to inform that he/she will be kicked if the server switches to a level requiring one of the missing expansion packs. The user can either choose to continue anyways or go buy the missing expansion pack.

- If the current level is not available to the client, an error along with the option to buy the missing expansion pack will be presented to the user.

Server region and country

A server is listed in a continent and country in the server browser. The default continent and country is Antarctica.

These settings are controlled by your RSP, and normally indicate the region & country where the server is physically located. The RSP has final say over the setting, but you can contact your RSP and ask to have the server moved to a different country. The region will change automatically when changing country.

Server pingsite

There are a half-dozen pingsite locations around the world. When a player initiates a Quickmatch operation, the player's PC will measure latency to all the pingsite locations.

Servers specify explicitly which of the pingsites is closest to them. The setting is controlled by the RSP.

Quickmatch combines the player's pingsite latency information with the server's pingsite setting when deciding whether or not to route a player to that specific server.