BFH PC Server Administration

This is the server admin manual for BFH PC Server R2.

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/*.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", which can be used to connect to the remote administration interface. Once connected, there is an assortment of commands available that can be sent. See "BFH 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 BFH 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 three 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**)
- During runtime the game engine will not allow the desired max-cap to be set to anything lower than the current number of players
- Depending on game mode there may also be a commander on each team which are included in the maximum amount of players

The current desired max number of players is the minimum of the RSP's max-cap and the **vars.maxPlayers** value.
On UNRANKED and PRIVATE servers, the engine will accept runtime changes to the setting within a second. However, if the **vars.maxPlayers** setting is set to anything lower than the current number of players on the server, the effective max number of players on the server will remain unchanged. The server will then retry changing the effective max-cap every 10 seconds until it succeeds.

So, in order to reduce the max number of players on a running server, first change the **vars.maxPlayers** setting and then manually kick people as necessary to get the current playercount low enough. The server will not kick players on its own accord.

vars.hacker can be used to allow or disallow Hacker on the server.

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

BFH 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 game modes 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 map list 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:

```
levels/mp/mp_bank TurfWarLarge0 1
levels/mp/mp_glades Heist0 1
levels/mp/mp_downtown Bloodmoney0 1
levels/mp/mp_bloodout Hit0 1
levels/mp/mp_desert05 Hotwire0 1
```

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-round

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.

Server types

When a server starts up, it will default to being Official. You can use **vars.serverType <type>** to make it any other server type during startup. The **vars.serverType** command is best placed at the top of Admin/Startup.txt. If the server is to be set to official, an additional command (**vars.mpExperience**) needs to be put in the startup. All official servers need a set experience to run. Also, please note that the mp experience will override any existing map list in the admin folder. Note that in Official servers the experience will be cleared if the server has a player count different than the default.

Ranked and Official servers will automatically run with PunkBuster and FairFight enabled.

Below is a list of ranges different server settings can be put to depending on server type.

Setting	OFFICIAL	RANKED	UNRANKED
vars.friendlyFire	FALSE	TRUE/FALSE	TRUE/FALSE
vars.IdleTimeout	300	225-86400	30-86400
vars.autoBalance	TRUE	TRUE	TRUE/FALSE
vars.teamKillCountForKick	5	4-10	1-99
vars.teamKillKickForBan	3	3-10	1-99
vars.vehicleSpawnAllowed	TRUE	TRUE/FALSE	TRUE/FALSE
vars.regenerateHealth	TRUE	TRUE/FALSE	TRUE/FALSE
vars.onlySquadLeaderSpawn	FALSE	TRUE/FALSE	TRUE/FALSE
vars.minimap	TRUE	TRUE/FALSE	TRUE/FALSE
vars.hud	TRUE	TRUE/FALSE	TRUE/FALSE
vars.miniMapSpotting	TRUE	TRUE/FALSE	TRUE/FALSE
vars.3dSpotting	TRUE	TRUE/FALSE	TRUE/FALSE
vars.killCam	TRUE	TRUE/FALSE	TRUE/FALSE
vars.3pCam	TRUE	TRUE/FALSE	TRUE/FALSE
vars.nameTag	TRUE	TRUE/FALSE	TRUE/FALSE
vars.hitIndicatorsEnabled	TRUE	TRUE/FALSE	TRUE/FALSE
vars.playerRespawnTime	100%	75-125%	1-300%
vars.soldierHealth	100%	60-125%	1-300%
vars.bulletDamage	100%	75-125%	1-300%
vars.forceReloadWholeMags	FALSE	TRUE/FALSE	TRUE/FALSE
vars.roundStartPlayerCount	4-8	4-8	1-10
vars.hacker	TRUE	TRUE/FALSE	TRUE/FALSE
vars.gameModeCounter	100%	75-400%	1-500%
vars.vechicleSpawnDelay	100%	25-400%	1-500%
vars.ticketBleedRate	100%	75-125%	1-300%
vars.roundTimeLimit	100%	50-300%	0-900%
vars.roundStartReadyPlayersPercent	35%	0-50%	0-100%
vars.teamSwitchingAllowed	TRUE	TRUE/FALSE	TRUE/FALSE
vars.requireReadyPlayersToStart	FALSE	TRUE/FALSE	TRUE/FALSE
vars.roundWarmupTimeout	45	30-60	5-90
vars.killFeed	TRUE	TRUE/FALSE	TRUE/FALSE
vars.roundsToWin	5	4-7	2-100
vars.roundStartReadyPlayersPercentRoundBased	70-70	40-100	0-100
vars.maxPlayer	10-64	10-64	0-64
vars.maxSpectator	0-4	0-4	0-4

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.preset – used to set the server preset: NORMAL, HARDCORE, or CUSTOM. You can also set the argument <lockPresetSetting> to true. This will override any settings that conflicts with the preset.

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. Only works on Unranked servers.

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.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

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

vars.playerRespawnTime – controls player respawn delay; specified in percent

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.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).

vars.ticketBleedRate – controls the rate of which tickets drop when the majority of control points are held by a team (100% = normal).

vars.roundTimeLimit – controls the amount of time a round will play (100% = normal).

vars.hacker – When set hacker is enabled.

vars.roundStartReadyPlayersPercent – Sets how many players are required to be ready before the

round starts

vars.teamSwitchingAllowed – Controls if team switching is allowed or not

vars.roundWarmupTimeout – Controls how long the warmup timeout is

vars.killFeed – When set, the kill feed is enabled

vars.roundsToWin - Sets the amount of rounds required to win (Crosshair/Hostage), takes effect at the end of the round

vars.maxSpectator – Sets the server max spectator cap

vars.maxPlayer – Sets the servers max player cap

vars.roundStartReadyPlayersPercentRoundBased – Sets how many players are required to be ready before the round starts for the round based game modes (Crosshair and Rescue)

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 official 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.

The serverinfo command reports if your server has matchmaking enabled or not, look at the serverinfo documentation for more information.

In addition to being matchmakable, your server will also need to have at least one player on it for the matchmaking system to send any quickmatching players to your server.

Starting with the patch released on April 28th 2015, in an effort to provide a more consistent quickmatch experience, Official Normal servers have to have the default player count for the mode to be eligible for quickmatch, to make sure that this is the case use the startup.txt file by specifying the type and mode, max number of players, and following the experience of the server using 'vars.mpExperience':

```
vars.serverType OFFICIAL
vars.preset NORMAL
vars.maxplayers <numplayers>
vars.mpExperience <Experience>
```

i.e.:

```
vars.serverType OFFICIAL
vars.preset NORMAL
vars.maxplayers 32
vars.mpExperience HST_VAN
```

Additionally if the server limits the player count using -numslots it will need to be higher or equal than the default player count for the experience to be recorded and applied. For example it won't be possible to host an Official Normal Hotwire server if the -numslots is set to 26 but it will work if it is

50.

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 shows all servers regardless of any game mode-affecting settings). A note on presets is that they are predetermined server setups and cannot be changed. If a setting that is included in a preset is changed then the preset will always go to “Custom”.

Setting	Normal	Hardcore
vars.friendlyFire	FALSE	TRUE
vars.IdleTimeout	300	300
vars.autoBalance	TRUE	TRUE
vars.teamKillCountForKick	5	5
vars.teamKillKickForBan	3	3
vars.vehicleSpawnAllowed	TRUE	TRUE
vars.regenerateHealth	TRUE	FALSE
vars.onlySquadLeaderSpawn	FALSE	TRUE
vars.minimap	TRUE	FALSE
vars.hud	TRUE	FALSE
vars.miniMapSpotting	TRUE	TRUE
vars.3dSpotting	TRUE	FALSE
vars.killCam	TRUE	FALSE
vars.3pCam	TRUE	FALSE
vars.nameTag	TRUE	FALSE
vars.hitIndicatorsEnabled	TRUE	FALSE
vars.playerRespawnTime	100%	100%
vars.soldierHealth	100%	100%
vars.bulletDamage	100%	100%
vars.forceReloadWholeMags	FALSE	TRUE
vars.roundStartPlayerCount	4	4
vars.ticketBleedRate	100%	100%
vars.hacker	TRUE	FALSE
vars.gameModeCounter	100%	100%
vars.vechicleSpawnDelay	100%	100%
vars.roundTimeLimit	100%	100%
vars.roundStartReadyPlayersPercent	35%	35%
Vars.teamSwitchingAllowed	TRUE	TRUE
vars.requireReadyPlayersToStart	FALSE	FALSE
vars.roundWarmupTimeout	45	45
vars.killfeed	TRUE	FALSE
vars.roundsToWin	5	5
vars.roundStartReadyPlayersPercentRoundBased	70	70
vars.maxSpectator	4	4
vars.maxPlayer	64	64

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.

Game queue / VIP (Reserved Slots)

If a game server is full people who join will be placed in a queue. There are some options to control how people are added from the queue.

Priority order:

1) VIP players (players added to the reservedSlotsList)
2) Premium players (players who have bought Premium)
3) Normal players.

If players are added to the reservedSlotsList the will be moved in ahead of normal and Premium players. If **reservedSlotsList.aggressiveJoin** is set to true and a VIP enters the queue the server will kick a non-VIP from the game to make room for the queued VIP. The server will first try to kick a non-VIP player who is not in a group and first if no such player could be found will it kick a grouped non-VIP. When kicking a players we do not make any distinguish between Premium and Normal players. A normal or Premium player joining the queue will never trigger a kick.

ReservedSlotsList.txt format

Each line should have one player name. There can be a maximum of 500 entries in the list.

Spectator list

If the game server is set to a server type lower than official, the spectator list can come in to use. On official, **vars.alwaysAllowSpectators** is set to true by default. If the server is hosted with another server type then this command can be set to false. If this is the case, then only players that are in the spectatorList.txt can join the server as a spectator.

For information on how to use the **spectatorList.*** commands, consult "BFH PC Remote Administration Protocol.pdf