

MoHW PC Server Provider Manual

Draft 17 October, 2012

Installation and maintenance of MoHW PC Server R1.

Terminology

Server	A game server, or game server instance
Client	A game client, i.e. a player of the game
Host	A physical or virtual PC that runs the game server instance
Blaze	EA's online backend. It handles player accounts, tracks servers etc. Any game clients and game servers need to connect to Blaze before they can find each other.
Server provider	A person or a company who maintains game servers and game server instances
Server administrator	A person who has limited, non-direct access to a game server instance; he/she has control over gameplay related aspects only

Runtime environment

Recommended OS: Windows Server 2008 R2.

The server has tested to work under:

Windows Server 2008 R2 (64-bit)

Windows Web Server 2008 (64-bit)

Windows 7 (64-bit)

The server software will **not** work on Windows Server 2003, Windows Vista, or Windows XP.

Connectivity

Inbound traffic

The game server will be listening for incoming traffic on the following ports:

Port	Type	Used by	Description
Set by RSP	UDP	Game Engine	Game clients connect to the game server on this port. Port number specified via "-port" commandline option. Default is 25200.
Set by RSP	TCP	Remote administration	Command port for the remote administration interface.

		Port number specified via “-RemoteAdminPort” option. Default is 47200.
ICMP	Game client’s server browser	Enable this, so the game clients can measure ping to your server. If this is blocked, your server will have high or nonexistent ping in the game client’s server browser.

Outbound traffic

The game server will perform outbound traffic to the following locations:

IP:Port	Type	Description
159.153.235.22:42100-42199	TCP	Blaze redirector
92.52.124.106:10000-20000	TCP	Blaze traffic
92.52.124.107:10000-20000		
92.52.124.108:10000-20000		
92.52.124.109:10000-20000		
92.52.124.102:10000-20000		
92.52.124.103:10000-20000		
92.52.124.104:10000-20000		
5.79.9.132:10000-20000		
92.52.124.87:10000-20000		
92.52.124.88:10000-20000		
92.52.124.89:10000-20000		
92.52.124.90:10000-20000		
92.52.124.101:10000-20000		
gostelemetry.blaze3.ea.com:9988	TCP	Blaze traffic
reports.tools.gos.ea.com:80	TCP	Blaze traffic
master0.evenbalance.com:14134	UDP	PunkBuster traffic
master1.evenbalance.com:14134		
master2.evenbalance.com:14134		
master3.evenbalance.com:14134		

master4.evenbalance.com:14134

wf2.evenbalance.com:14134

wf1.evenbalance.com:14134

wf-b.evenbalance.com:24340

namereg.evenbalance.com:24360

Connections to the backend

To ensure a stable server, you need to be able to run a stable TCP connection to 92.52.124.*. Here is why:

The game server will establish a TCP connection to a randomly chosen machine in the 92.52.124.* range during startup. Those machines are located in a datacenter in the London area. This TCP connection is used by the game server for most communications with the Blaze backend.

The TCP connection is required to be up and stable for the server to be present on Battlelog and for players to be able to play on the server.

If the TCP connection unexpectedly breaks due to any sort of networking problem, the game server will attempt to reconnect over a period of 30 seconds. Once the grace period is out, the game server will disappear from Battlelog and any players present on the server will be ejected from the server.

If the TCP connection is down, the server will attempt to reconnect to the Blaze backend with regular (but increasing) intervals.

Connections to game clients

Traffic to game clients is purely UDP-based. The server and client accept that traffic stops for a few seconds, but if there is no communications for longer than that, the server will consider the connection with the client severed (and vice versa) and will kick it out.

Quick start guide

Add your machines to the IP whitelist

- Report which IP ranges that you will run servers on to Danger Close. Do this at an early stage, because IP whitelist configuration takes a long time. See section “IP whitelist configuration” for details.

Get connected with the RSP community

- Each RSP will automatically be added to the Danger Close Games RSP mailing list at mohw@lists.multiplay.co.uk. If you wish to add more users you may do so by mailing btingle@ea.com and asking to be added.

Acquiring materials

- Get the game server (approximately 1.2GB) from DCG. Sometimes it is just one archive, sometimes it consists of several updates.
- Get a “ProviderId.dat” file from DICE. You need this one in order to start any server instances.

Installation

- Unpack the server archive to a directory on the host PC. The server will need read+write rights to this directory.
- Deploy the “ProviderId.dat” file to the installation directory.
- Note: the server should have all the system DLL files you need to run, but if not you may need to install the MSVC 2008 or DirectX Redistributable package. Should a game server be unable to start, please contact Danger Close Games and we will assist you.

Firewall configuration

- Open necessary port in your local firewall, according to the table in the “Connectivity” chapter. Incoming ports need to have unique port numbers for each instance.

Configuration of server instances

- Make one copy of the ExampleServerConfiguration directory per server instance.
- Configure each server instance according to the “Offline configuration” chapter.

Running server instances

- Launch the server executable (MOHW_server_Final.exe) multiple times – once per instance – with proper arguments at least for “-**ServerInstancePath**”, and a unique name & port. Name & port can be configured either via commandline or via ServerOptions.cfg.

Monitoring the server instances

- Ensure that the server instances have unique port numbers for their remote admin ports.

- Regularly connect to the remote administration port of each instance (see “Remote Administration interface” chapter for details) and issue a “serverInfo” command. The response contains some brief info on what the server is doing at the moment.

Server executable name

The name of the main server executable is “MOHW_server_Final.exe”.

Server instances

One server installation can support multiple server instances.

Preparation

Each server instance needs its own home directory, with read+write rights.

These files can exist in the instance’s home directory:

ServerOptions.cfg

ProviderId.dat

Admin/MapList.txt

Admin/Startup.txt

If PunkBuster is enabled on the server, it will create a directory named “pb” there and store instance-specific files in there.

If RuntimeLogs are enabled, then the server will create log files in the Logs/ subdirectory.

Execution

Start the server executable multiple times, each time letting the “-ServerInstancePath” commandline parameter point out a different home directory that you created above.

IP whitelist

Before a server can communicate with the Blaze backend, its IP address must be added to an IP whitelist. DCG/EA manage this whitelist.

You add entries to the IP whitelist by mailing a list of the IP ranges that you control to both dstanfill@ea.com and btingle@ea.com.

IP ranges can be specified in two different formats:

- 1) Start IP – end IP; example: 192.168.0.27 – 192.168.0.119
- 2) CIDR format; example: 192.168.0.0/25

Typical turnaround time for IP whitelist changes is 3 business days, but it can take up to 10 business days. Due to the long turnaround times, it is better if you don't submit one IP at a time, but rather submit broader ranges (which include machines of yours where you might be running servers in the future).

Configuration options

Configuration options are divided into three categories:

- Early startup settings, only controllable by the server provider
- Options which should only be controllable by the server provider
- Options which may be controlled either by the server provider or the server operator

Early startup settings (commandline)

-ServerInstancePath <path> (default: <local user>\Documents\Server)

Specifies where the instance-specific files are kept.

-Game.LogFileCollisionMode <mode> (default: LFCM_Rotate)

Each time that the game server is started, a new RuntimeLog is created. This setting specifies what should happen with older logs.

Allowed values for this setting:

Value	Interpretation
LFCM_Overwrite	Overwrite previous log
LFCM_Rotate	Rename all older logs; keep max 20 logs
LFCM_TimeStamp	Put a unique timestamp in each log name

Server provider configuration options (commandline, ServerOptions.cfg)

These are any options that should only be available to the server provider. Those usually have something to do with the network architecture or the on-disk structure of the server machine. They are all on the form of, -<option> <value>. Example:

-GamePort 25200

These options can be specified either on the command line or in the ServerOptions.cfg file. The priority is:

- command line
- ServerOptions.cfg file

Basic configuration

-GamePort <port> (default: 25200)

Port to use for game UDP traffic; game clients will attempt to connect to the server via this port. The game server will listen to this port on all interfaces.

-GameIP <ip> (default: not set)

Specifies which network interface should be used for game traffic. If this is not set, the game will listen on all network interfaces in the machine.

-RemoteAdminPort <port|ip:port> (default: not set)

If set, enables the remote administration interface on the specific port. If you do not specify an IP, the game server will only listen for connections from the local machine. Specify 0.0.0.0 as IP if you wish to listen on all network interfaces.

-NumSlots <count> (default: 64)

Number of players allowed onto the server at the same time. The server administrator can lower the number of players further by using the **vars.maxPlayerCount** remote administration command.

Multithreading

The game server is (mostly) singlethreaded by default. If your server machine does not have a high enough CPU frequency, you can enable multithreading to get around this limitation.

-Server.ThreadingEnable <enabled : boolean> (default: false)

By default threading is disabled. If set to true, threading will be enabled; then, the number of processors that will be used is controlled by -Core.JobProcessorCount.

-Core.JobProcessorCount <numCores : integer>

The default for a server instance – assuming that threading is enabled – is to use all cores on the machine. You can use this setting to limit the number of cores that the server instance will use. A sensible number is somewhere in the 2 – 4 cores area; increasing the parallelization above that amount will make the server instance burn more resources without running much faster.

Server location

-Region <region> (default: AC)

Normally specifies where in the world the server is physically located. The game client has a region-filter, which filters servers depending on this setting.

Allowed values for this setting:

Value	Interpretation
NAm	North America
SAm	South America
AC	Antarctica
Afr	Africa
EU	Europe
Asia	Asia
OC	Oceania

-Country <country> (default: AQ)

Normally specifies where in the world the server is physically located.

Allowed values for this setting are the Alpha-2 codes in the ISO 3166-1 standard:

http://en.wikipedia.org/wiki/ISO_3166-1

The country setting is controlled by the server provider. Server administrators shall not be able to control the country setting directly, but can ask the server provider to have it changed. If the country is changed, then the region should be changed accordingly.

-PingSite <location> (default: not set)

Specifies which “ping site” is nearest to the game server. This is used as a hint by the Quickmatch operation in the game and on Battlelog, which attempts to funnel players to servers in their vicinity.

Allowed values for location:

location	Corresponding region
mis-lax	Western US
mis-iad	Eastern US
i3d-ams	Europe
i3d-syd	Australia
i3d-nrt	Japan

Sets the name of the server. This is the name that will be displayed in Battlelog.

+admin.password <password>

Specifies the password for the remote administration interface. If this is not set, it is not possible to log in to the remote administration interface – and so only a limited set of commands will be available via that interface.

+vars.gamePassword <password> (default: none)

If set, the server will have a game password. Any players which wish to play on the server will be required to give the right password before they are allowed to enter the server.

There are many more commands available – see the Remote Administration Protocol specification for details.

Troubleshooting

Are your servers acting up? I'm starting to be surprised. Here are some tips along the way.

On your own

Run a single server, and dig through the RuntimeLog as described in the "Reading RuntimeLogs" section.

Reporting issues to Danger Close Games

First, reproduce the problem on one of your servers.

Then, make this available to DCG on your FTP:

- Appropriate RuntimeLogs
- Appropriate CrashDumps

And then email DCG, dstanfill@ea.com and btingle@ea.com.