Plains of Pain / Dedicated Server / Windows Setup

Updated: June 12, 2024 by Cobra Byte Digital

Common Requirements

- reserve 50-250 kbps of bandwidth per 1 player

- (default) allow/forward **UDP** ports **7777** and **27016**

- installed SteamCMD

- 400 MB free space for installation

Performance Requirements Overview

- 4 GB RAM and 2 CPUs (for small world)

- 8 GB RAM and 4-6 CPUs (for large world)

- 16-32 GB RAM, 8-16 CPUs for 200 players on any size world

Installation

- Install **SteamCMD**

  - There is an official Steam documentation here:
    https://developer.valvesoftware.com/wiki/SteamCMD#Windows

- Run **SteamCMD** and download the Plains of Pain (Dedicated Server) using following command:

  ```
  app_update 2227360 validate
  ```

Server Setup

Server can be configured with following parameters. Some of them are mandatory for booting at all, keep it in mind.

Server can be configured via CLI arguments or via JSON file.

Parameters

Server will not boot properly if mandatory parameters are missing.

Parameter	Default Value	Description	CLI	JSON
`startServer`	*empty*	Without this parameters server will NOT boot	mandatory	not available
`localhost`	*empty*	Without this parameters server will NOT boot	mandatory	not available
`nographics`	*empty*	Unity's default argument for avoiding GPU initializing	mandatory	not available
`batchmode`	*empty*	Unity's default argument for running app in headless mode	mandatory	not available
`ignorecompilererrors`	*empty*	Unity's default argument for ignoring certain compiling errors	mandatory	not available
`config`	*empty*	Invokes using JSON config. i.e. "default.json"	optional	not available
`serverName`	"My Server"	This name will appear in the server list in the game	optional	mandatory

Parameter	Default Value	Description	CLI	JSON
`maxPlayers`	10	Currently there is a limit up to 200 players	optional	mandatory
`port`	7777	Game port	optional	mandatory
`queryPort`	27016	Query port	optional	mandatory
`worldId`	0	Selected World ID, should be **1** on a new created server *	optional	mandatory
`seed`	40377	Seed for world generator	optional	mandatory
`mapId`	0	0 = Wasteland v0.4 1 = Dunes v0.4 2 = Dunes v0.5	optional	mandatory
`difficulty`	2	0 = Tourist 1 = Rookies 2 = True Wastelander 3 = Veteran 4 = Overlord	optional	mandatory
`worldSize`	5	3 = S 5 = M 7 = L 9 = XL 11 = XXL	optional	mandatory
`cc`	*empty*	Enables console commands if presents	optional	not available

(*) World IDs are created incrementaly. That means each time you want to boot World with non-existing ID, a new world is created and its ID is print to console.

Configuration via JSON

Server can be configured also via JSON files. The `configs/default.json` looks like this:

```
{
    "serverName":"My Server",
    "worldId":0,
    "mapId":0,
    "seed":40377,
    "difficulty":2,
    "worldSize":5,
    "port":7777,
    "queryPort":27016,
    "maxPlayers":10
}
```

Then the server can be executed from `cmd` like this:

```
"PlainsOfPain.exe" -nographics -batchmode -localhost -startServe
r -ignorecompilererrors -config "configs/default.json"
```

Server files are delivered with folder/file `configs/custom.json` which should be your configuration point.

There is a file `run_server_template_2.bat` in root folder which contains those arguments, so you should make a copy and use it.

Configuration via CLI

There is a default template file `run_server_template_1.bat` in the root folder of the downloaded server, it contains something like this:

```
"PlainsOfPain.exe" -nographics -batchmode -localhost -startServe
r -ignorecompilererrors -maxPlayers 100 -serverName "My Server N
ame" -worldId 1 -seed 12345 -worldSize 5 -mapId 0
```

You should make a copy (i.e. `run_server.bat`) and setup desired parameters there.

Then open Windows's `cmd` tool, locate server's root folder, and simply write and execute this:

```
run_server.bat
```

Server will boot.

Warnings

Those default files like `configs/default.json` , `configs/custom.json` , `run_server_template_1.bat` and `run_server_template_2.bat` could be overwritten when server files are updated, so we encourage you to make your own local files and use them.

Troubleshoot

Server should be delivered with this library, but if not, you can download it by yourself:

https://docs.microsoft.com/en-US/cpp/windows/latest-supported-vc-redist?view=msvc-170