MultiFPS - Setup server list

Contents:

Introduction

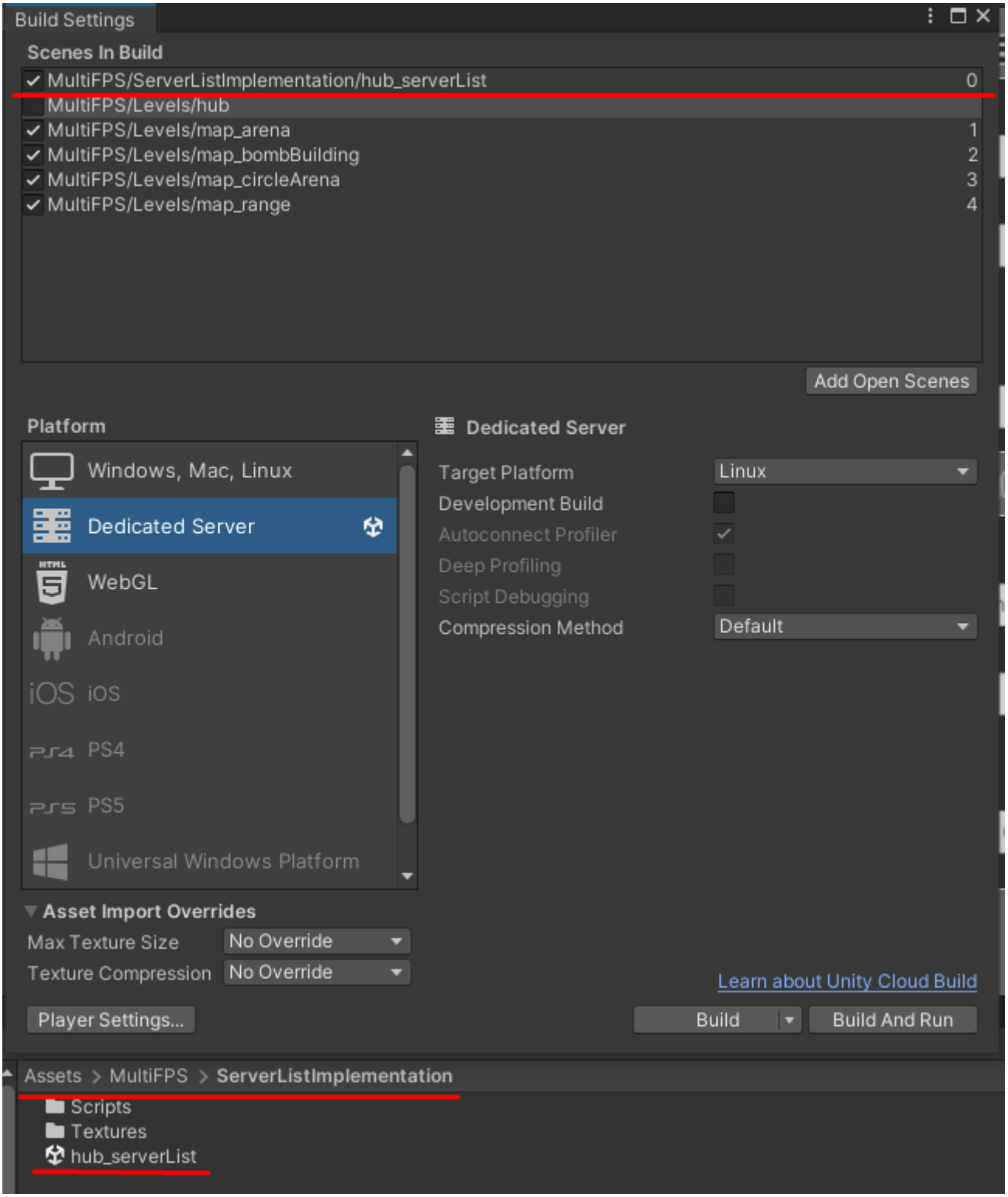
In this document we will go through process of setting up server list on Your own dedicated machine that game client (MultiFPS) will connect to in order to create games or join existing ones.

This manual assumes You can ssh to Your machine and transfer files to it.

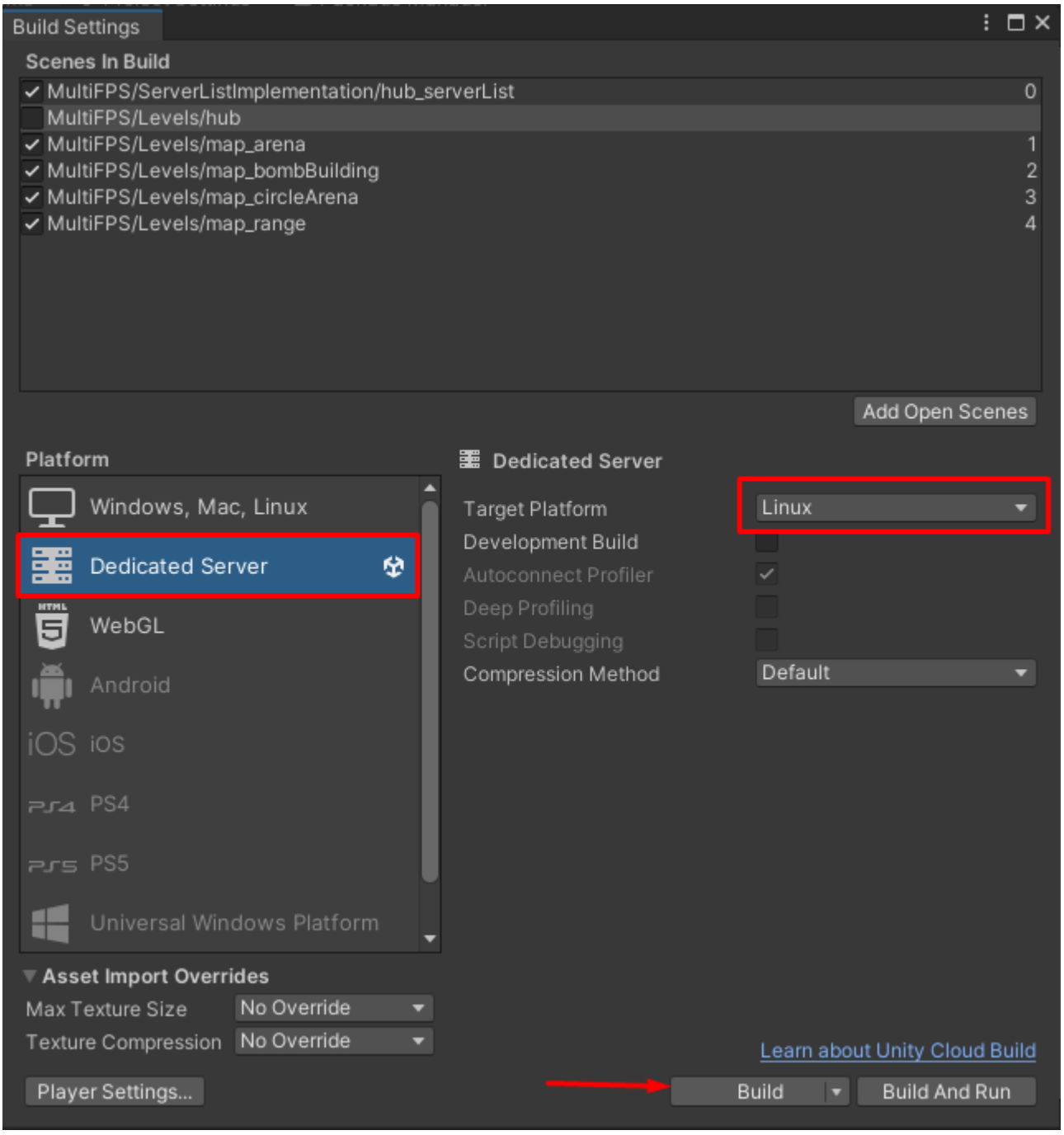
Setup Game Client

You must first set up a unity project as shown in the "MultiFPS Tutorial - Setup project.pdf" file included in this package, before applying these steps.

Load scene "hub_serverList" and add it to build settings. Make sure it have index 0, we're replacing the hub scene with this scene. It contains all the UI and code necessary to operate with server list that we'll setup later.



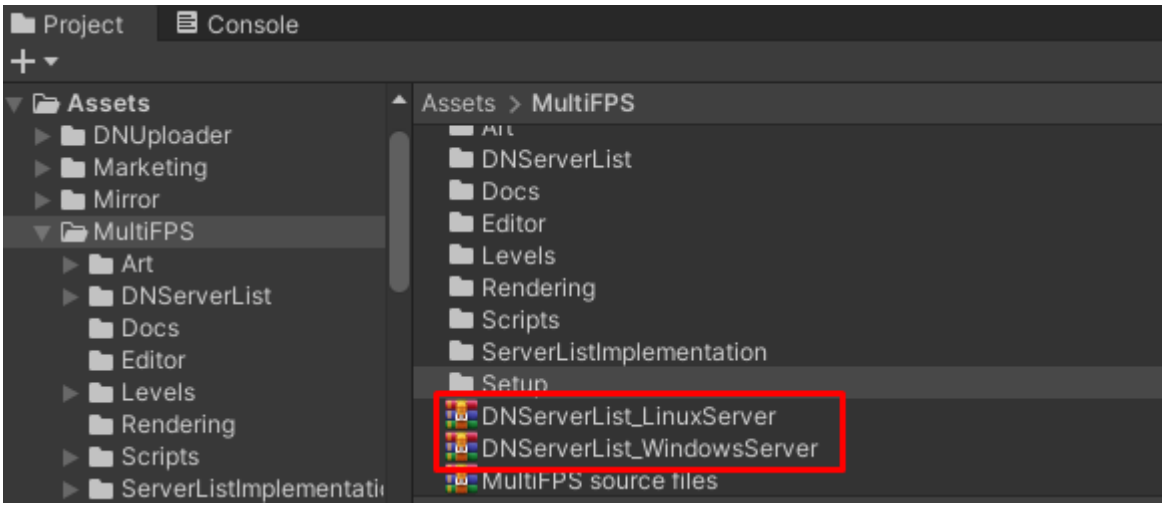
Build server player



Build a server player for linux by choosing those options and clicking the “Build” button. We will upload this build to the server alongside the server list app in the next part of this document.

Setup for linux server

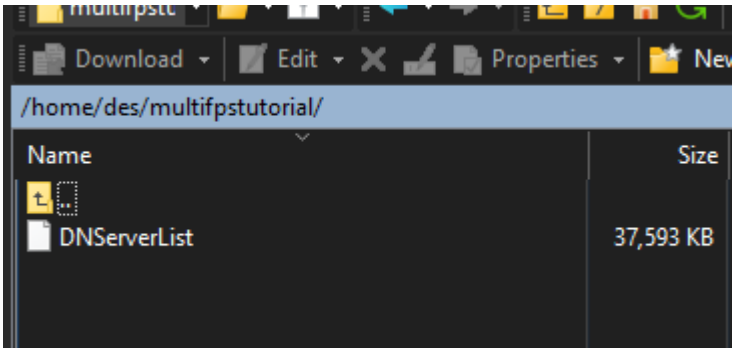
1. Transfer appropriate server app to Your machine. Inside both archives You'll find one file for a given system.



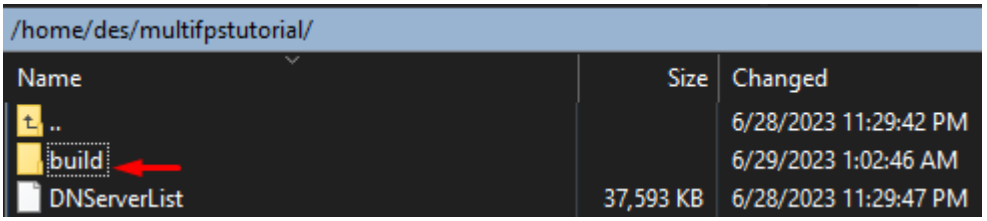
For this manual we will go with Linux.
For transfering file to server we'll use WinSCP, but it can be any other FTP client

In this manual I will be connecting to my VM from Windows 10, so for file transfering I used WinSCP, and for command line Putty.

For linux server we need to transfer file from .zip archive "DNServerList_LinuxServer". For windows server send .exe file from "DNUploader_WindowsServer". Both these files can also be run locally for testing. You can also transfer config file included in this archive, but it will also be generated on first launch of this app



In the same directory create folder "build" and paste previously builded game to it



Next navigate from terminal to location where You transferred file, I'm using putty for it

```
root@desNetware-NY:/home/des/DNUploader# ls
DNUploader_Server
root@desNetware-NY:/home/des/DNUploader#
```

To run serverlist we first need to give it execution rights. To do this run command:
chmod +x ./DNServerList

```
root@desNetware-NY:~# cd /home/des/multifpstutorial
root@desNetware-NY:/home/des/multifpstutorial# ls
DNServerList
root@desNetware-NY:/home/des/multifpstutorial# chmod +x ./DNServerList
```

Run DNServerList by executing:

./DNServerList

Now close the program by Ctrl+C and open the config file, it will be located in the same directory as this app. For editing config file I will use nano

```
root@desNetware-NY:/home/des/multifpstutorial# ls
DNServerList  config.txt
root@desNetware-NY:/home/des/multifpstutorial# sudo nano config.txt
```

Configuration file for linux server app

First launch of DNServerList will create a config file with all available parameters to configure. We will go through each of them.

```
#Config for REST API
RestApiUrl_HTTP: http://localhost:5000
RestApiUrl_HTTPS: https://localhost:5001
CorsAllowedDomains: https://desnetware.net/demos/multifps,http://localhost:3000
UseCORS: true
AlwaysRedirectToHTTPS: true

#Config for unity game builds
ServerBuildManagerCommunicationPort: 8000
ServerGameBuildExecutablePath: build/game.x86_64
GameBuildStartingPort: 9000
MaxGamesAtOnce: 3
NetworkAddress: desnetware.net
UseReverseProxy: True
ReverseProxyStartingPort: 27000
```

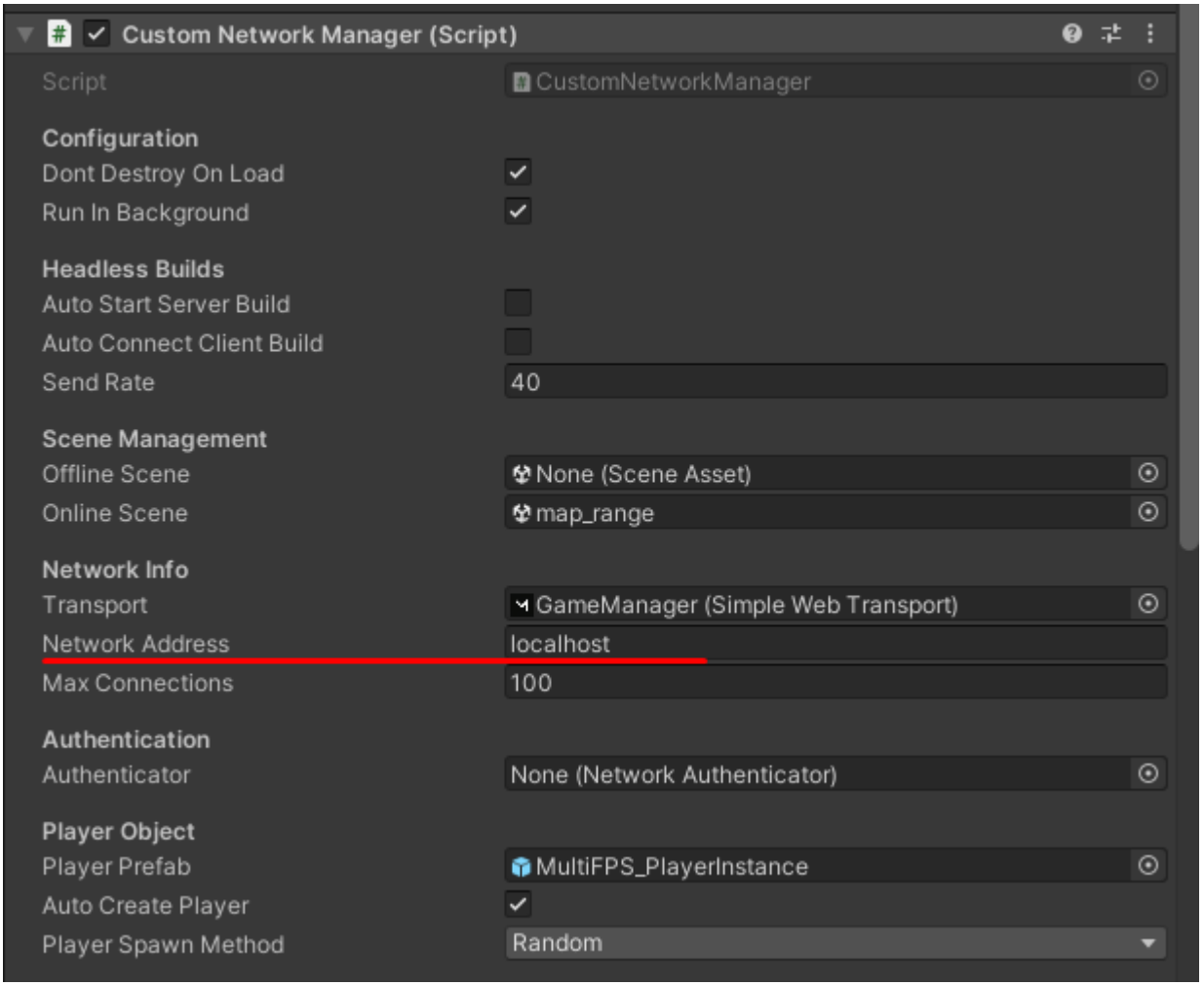
RestApiUrl_HTTP: on this url we will be sending http requests to server list app from unity client
RestApiUrl_HTTPS: same as above but for HTTPS protocol, must have if for example You'll be hosting Your game on site that uses https.

CorsAllowedDomains: domains that will be allowed by cors, You can have multiple of them but they must be provided in single line with comma separating them. For example I host demo of MultiFPS on my site on url "https://desnetware.net/demos/multifps" so I have to include it here
UseCors: disable or enable cors, set "true" or "false"

AlwaysRedirectToHTTPS: if true requests made on **RestApiUrl_HTTP** will be redirected to **RestApiUrl_HTTPS**

ServerBuildManagerCommunicationPort: On this port spawned server build instances will be communicating with server list app, so they can report state of the game to it, how many players are connected for example. No need to change it if You have this port free.
ServerGameBuildExecutablePath: path to unity server build with execution file
GameBuildStartingPort: from this port included unity server build will be listening on for incoming connections from players
MaxGamesAtOnce: how many games can be running at once

NetworkAddress: address of this machine that our server list app and games will be running on, this address will be sent to game client so it will be able to connect to our games. Game client will paste this address in Mirror's network manager.



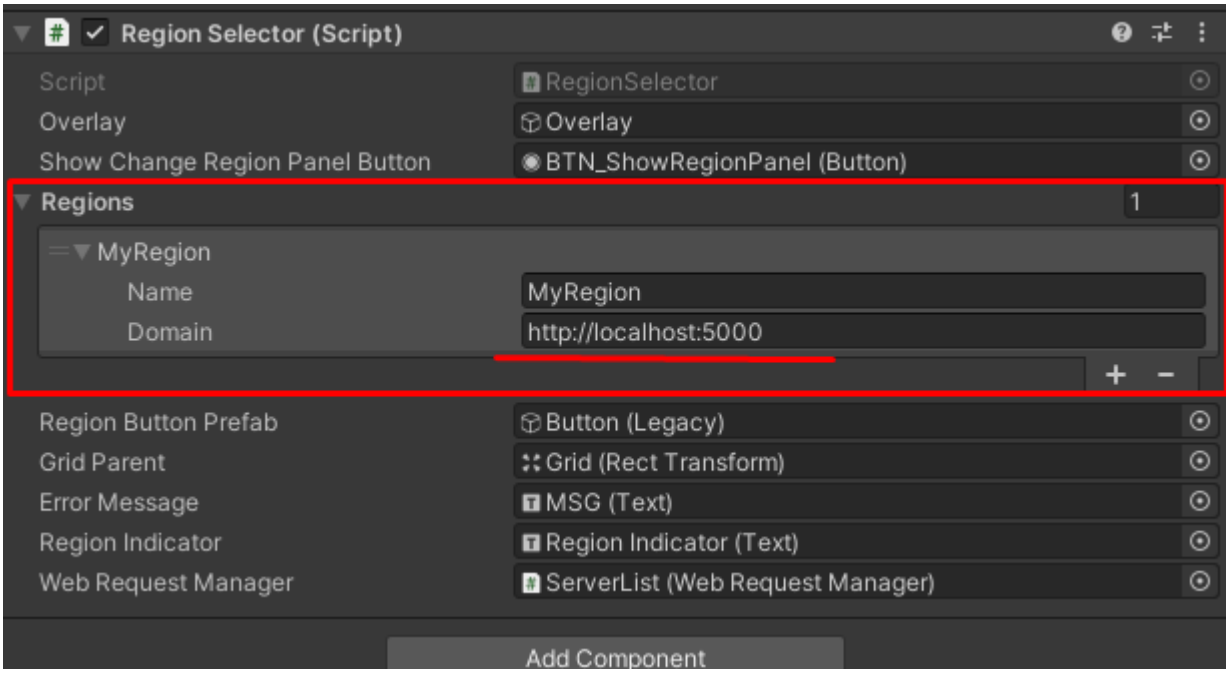
UseReverseProxy: Set to true if You want to put Your game servers behind reverse proxy, useful for hosting games for webgl clients hosted on sites that uses HTTPS, more on this topic specifically for mirror here:
https://mirror-networking.gitbook.io/docs/manual/transports/websockets-transport/reverse-proxy

ReverseProxyStartingPort: if **UseReverseProxy** is set to true, from this port included games will be hosted on, and should be accessed by players using ports from **GameBuildStartingPort**

So for example when config is set like this**:**
GameBuildStartingPort: 9000
ReverseProxyStartingPort: 27000
reverse proxy app should point port 9000 to 27000 on which game will be hosted, and for second game it will go like that: 9001 -> 27000,
and for third: 9002 -> 27002.

After ended configuration run Your app again and connect to it from unity client

In unity in scene “hub_serverList” that we added to build settings in first part of this document in object “RegionSelection” add Your custom region In field “Domain” paste url that You used in config file either in field **RestApiUrl_HTTP** or **RestApiUrl_HTTPS,** depending on which one You are using



Everything is set. You can now launch the game and use the server list from it.

Thank You for using MultiFPS
desNetware

Website
Contact
Youtube