# Survial Engine Online

Getting Started with Online Features

Thank you for downloading Survival Engine Online!

This document will explain how to setup your project and test the game's networking features.

If you are not familiar with offline survival engine, I really suggest to get familiar with the basic engine features before jumping into networking and multiplayer. There is another document explaining the offline concepts (items, crafting, constructions, plants…).

There are tutorial videos available on Youtube, and you may join the Discord server for any questions.

Youtube: https://www.youtube.com/channel/UC0LYig0AgPT9T5IN5DCUiqQ
Discord: https://discord.gg/JpVwUgG

If you still need help, you may contact me at: contact@indiemarc.com

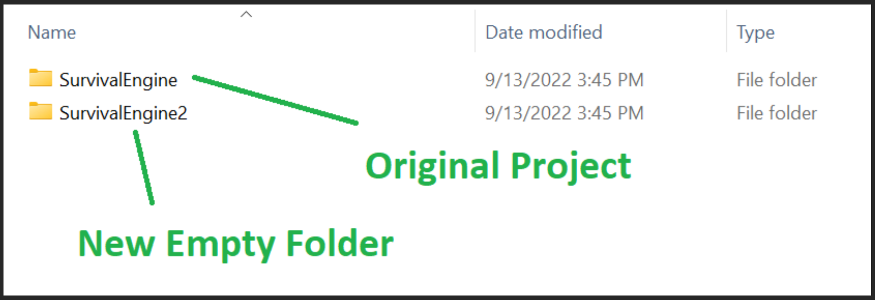
## Testing the game

The easiest way to test the game, right after downloading the asset, is to build the game client (with Menu as first scene). And then run it twice on your PC. You can Host with one, and Join the game on local ip (127.0.0.1) on the other one.

## Creating two Projects

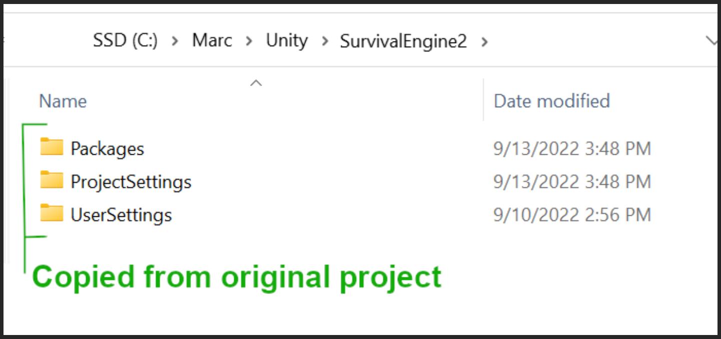
Even if creating a build is the simplest way to test the game, it is not the most efficient method. While working on your game, you may want to iterate quickly and test many times a feature, without having to create a new build for every single change. By default, Unity does not allow to open the same project twice, but there is a trick to allow this. It is probably possible to do it on Mac and Linux, but the following will explain how to do it on Windows:

Create a new empty project folder outside of the original project folder.



| Name | Date modified | Type |
|---|---|---|
| SurvivalEngine | 9/13/2022 3:45 PM | File folder |
| SurvivalEngine2 | 9/13/2022 3:45 PM | File folder |

Then, copy these folders from the original project to the new project: Packages, ProjectSettings, UserSettings (but not Assets)


SSD (C:) › Marc › Unity › SurvivalEngine2 ›
Name
Date modified
Packages 9/13/2022 3:48 PM
ProjectSettings 9/13/2022 3:48 PM
UserSettings 9/10/2022 2:56 PM
Copied from original project

Open the command prompt in admin mode (click on start, type "cmd" in the search, then right click on command prompt, and "run as admin".
Change to your new project folder by typing

```
cd C:\Marc\Unity\SurvivalEngine2
```

(replace with the path to your new folder)

Create a simlink to the Assets folder using this command

```
mklink /D Assets C:\Marc\Unity\SurvivalEngine\Assets
```

Replace by the path with the Assets folder of your original project. The /D options means it's a directory and not a file.

Done! You can now open both projects in Unity. Whenever you change any file inside Assets, it will automatically change it in the other project. The project settings won't be the same, but it allows you to setup the build settings differently in each project (could be one for the client, one for the game server).

From Unity, you can run the game directly from any scene (without going through the menu). Doing so will start the game as a "Host", which also includes singleplayer.

If you want to join another player, you need to go through the Menu, and click join.

## Network Types

There are 3 different network types you can use included in this asset. Each one have their own advantages.

**-Peer to peer (including LAN)**

**-Dedicated Server**

**-Relay Server**

### Peer to peer

Simplest way to connect and test since it doesn't require you to host a server. All you need to do do is Host a game from the Menu scene, and then Join with another player using the host's IP:

Local IP: **127.0.0.1** if running the game twice on the same PC.

Internal IP: **192.168.x.x** (this is LAN, you can see the host internal IP in the menu).

External IP: **x.x.x.x** (you can google "what is my IP" to know this).

The disadvantage of this method is that the host needs to open ports and allow the application on their firewall. Also, if connecting with external IP, the host will need to setup port forwarding on their router. The default port is 7700 but it can be changed in the NetworkData settings in Resources folder.

### Dedicated Game Server

A dedicated server means that there is no host, instead the host is a game server you upload on a cloud server, and run, that other people can connect to. This is a bit more complex to setup because you need to manage your own server.

There can only be 1 game per server app (although you can run the application multiple time on the same cloud server, each with a different port). To help with this, there is a lobby server (which is a separate build) that makes it easier for player to find other players, and that will automatically start and stop a new dedicated game server for each game room.

The advantage of this method is that there is no player host, so the game does not stop if the host leaves, and no one needs to setup port forwarding. But this method will require you to setup a server, and will require the most server processing power. Since the entire game logic runs on your server.

To build the dedicated server, put the ServerGame as the first scene, and make sure to also include all playable game scenes. Another document will explain how to install this.

For testing purposes, you can run the dedicated server locally on your pc, either from a build or from unity. You can also connect directly to a dedicated server in the same way you would connect to a host: from clicking on Join in the Menu and using the server's IP.

## Relay Server

Using a relay server can be used as a compromise between the other 2 methods. You will still have a host (so game shuts down if the host leaves), but the host won't need to setup port forwarding and do not need to setup their firewall. The relay server does not handle game logic, but will act as a middleman, transferring all messages between the client and the host.

A relay server will require way less processing power than a dedicated server, but it will have the greatest delays, since all messages needs to transit through the relay, and if the relay is far from your host it may increase network delays.

In this asset, the offered relay server solution is with Unity Services Relay. More info here: https://docs.unity.com/relay/

To use it, you will need to link your project to a Unity ID inside the Services Window, and you will need to enable Unity Relay on your Unity dashboard: https://dashboard.unity3d.com/

At the moment, in this asset, you can't connect directly to a relay server (since the process is a bit more complex). Instead, you need to connect through the lobby server in Relay mode.

## Lobby Server

The lobby server is a separate application that can help your players to find each other. Unlike the game server, the lobby server only needs to be run once, and as many players as you want can connect to it (as long as the server specs allows it). Once a player joins a game room and starts the game, it will disconnect from the lobby server and connect to the host or game server.

The lobby server can be used with any of the **3 network types** above. To change which network type the lobby uses, edit the Resources/NetworkData file.

The lobby server itself is always dedicated, but after the player connects to a game, it will connect to the appropriate host (either peer-to-peer to another player, or through the relay, or to a dedicated game server).

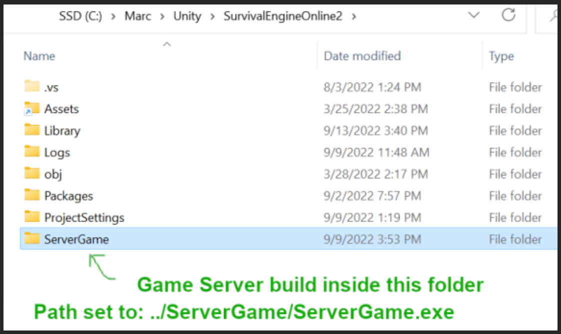
**Warning**: By default, in Resources/NetworkData, the lobby_url is set to server.survival-engine.com. This is a dedicated server I host myself, while it can be used for quick testing a new blank project containing unchanged Survival Engine, as soon as you start changing the code or add your own scenes or objects, it will break compatibility and you will need to host your own server.

### Dedicated Game Server Launcher

When the lobby is in dedicated server mode, the lobby will also take care of starting the dedicated game servers (one for each game, and each one with a different port). To make this work, you need to include the game server build inside a folder (the path can be set in NetworkData).

The path is absolute, unless it starts with ./ or ../
In that case the path is relative to unity's Application.dataPath, which is either the data folder of the build, or the assets folder of the unity project.

This example would allow you to run the lobby in dedicated mode from Unity:



SSD (C:) › Marc › Unity › SurvivalEngineOnline2 ›

| Name | Date modified | Type |
|---|---|---|
| .vs | 8/3/2022 1:24 PM | File folder |
| Assets | 3/25/2022 2:38 PM | File folder |
| Library | 9/13/2022 3:40 PM | File folder |
| Logs | 9/9/2022 11:48 AM | File folder |
| obj | 3/28/2022 2:17 PM | File folder |
| Packages | 9/2/2022 7:57 PM | File folder |
| ProjectSettings | 9/9/2022 1:19 PM | File folder |
| ServerGame | 9/9/2022 3:53 PM | File folder |

Game Server build inside this folder
Path set to: ../ServerGame/ServerGame.exe

### Build the Lobby

To build the lobby server, you only need to include one scene: ServerLobby.

For players to access to lobby, they can do it from this scene: MenuLobby

For online play, you will need to upload and run your lobby server executable on a cloud server. If in dedicated server mode, you need to do the same with the game server executable. More info on this in another document: Cloud Server Setup.