Tutorial: Creating a multiplayer game project

This tutorial walks you through the steps needed to create a multiplayer game project with Lumberyard. At the end of the tutorial you will have a new project and a basic understanding of how to connect two players on the same pc using the in-game console.

You will learn how to do the following:

- Use the Project Configurator
- Create a new game project
- Enable and disable Gems
- Compile the editor, engine and game
- Use the console to connect two players locally

Prerequisites

This tutorial assumes you have installed Lumberyard and set up your development environment.

Step 1: Create a new game project

This step in the tutorial will show you how to create a new game project using the Project Configurator.
To Create a New Game Project

1. Run the Project Configurator from the Start Menu, or run ProjectConfigurator.exe from the dev/Bin64 folder

The Project Configurator will launch and display the list of projects.
2. Click the **New project** button and the **Create a new project** dialog will open.

3. Enter a clever name for your new project and press **Create project**.
4. Select your project by clicking on it, and then press the **Set as default** button to make it the active project.

At this point you could close the configurator, compile your project and run it. The default project has all the networking functionality you need for a multiplayer game enabled by default, however, in this tutorial we'll enable the Multiplayer gem to add additional Flowgraph nodes.

**Step 2: Enable the Multiplayer and UserLoginDefault gems**

This step in the tutorial will show you how to enable the Multiplayer and UserLoginDefault gems in your new project. The Multiplayer gem is light-weight and contains useful Flowgraph nodes for listing servers, joining servers and more. We'll be using the Multiplayer gem to easily create logic for a simple UI. The UserLoginDefault gem provides a default user profile which is a requirement if we want to distribute our game client or server and run them in release mode.
To enable the Multiplayer and UserLoginDefault gems

1. Press the Enable package settings button in the header to see Gems settings for your new project.

Gems are optional functionality and assets you can add to your game project.
2. Scroll till you see the Multiplayer gem then select the checkbox for it.

3. Next, enable the UserLoginDefault gem by selecting the checkbox next to it.
4. Press the **Save** button in the header to apply these settings.

![Project Configurator screenshot](image)

5. Close the Project Configurator.

Great! You now have all the gems enabled that we'll be using for this tutorial series.

**Step 3: Compiling your new project**

This step in the tutorial will show you how to configure and compile your project from the command line using lmbr_waf.

**To configure and compile**

1. In a command window, navigate to the dev folder inside your Lumberyard folder. Example: If you installed Lumberyard in C:\Amazon\Lumberyard\1.1.0.0 navigate to C:\Amazon\Lumberyard\1.1.0.0\dev
2. Run `lmbr_waf configure`

If you see any error messages about missing required 3rd party software, run SetupAssistant (formerly known as LumberyardLauncher in V1.1 and under) and install the required software before continuing.
3. Run `lmbr_waf build_win_x64_profile -p all`

After the compile process finishes successfully, you are ready to run the editor to make a new level for your game.

**Step 4: Connecting players using the in-game console**

Even before you have any levels, menus or gameplay set up, you can connect players using the game console. This step will show you how to run two game clients on the same pc and connect them using console commands.

**To connect two players using the game console**

1. Navigate to the dev/Bin64 folder and run the launcher executable for your game. For example, if your game is named TutorialProject, run `TutorialProjectLauncher.exe` from the dev/Bin64 folder.
The game client will open and display a black window.

2. Press the ` key (just above the TAB key) to open the console. Run the command `mphost` to make this client host a network session other players can connect to.
3. Start another game client by running your launcher while the first one is still open. For example, if your game is named TutorialProject, run TutorialProjectLauncher.exe from the dev/Bin64 folder.

The new game client will open and display a black window.

**HINT:** If you want to arrange the two game client windows so you can see them at the same time, press the ` key to open the console, and while the console is open use the mouse to drag the window by clicking and dragging the title bar.
4. In the new game client window, press the ` key (just above the TAB key) to open the console. Run the command **sv_port 0** to set this client's network port to an ephemeral port. If you tried to connect without changing the network port the connection would fail because the first client is already using the default port.
5. Next, run the command **mpjoin** to join the first game client’s network session.

You should see in the console logs that the client connects to the hosting game client.
6. Switch to the first game client by pressing **ALT+TAB**. Run the command `gm_debugdraw 1` to turn on a debug view with basic network information.

![Image of game window with debug information]

Changing the value of the **gm_debugdraw** console variable will change what network debug information is displayed. The information displayed is controlled by bit flags as follows:

- **Basic** = BIT(0)
- **Trace** = BIT(1)
- **Stats** = BIT(2)
- **Replicas** = BIT(3)
- **Actors** = BIT(4)
- **EntityDetail** = BIT(5)
- **Full** = **Basic** | **Trace** | **Stats** | **Replicas** | **Actors** | **EntityDetail**

The console command `gm_setdebugdraw` will set **gm_debugdraw** to 31 and display all information.
7. Press the ` key to close the console so you can see the network debug view.

The network debug view shows that this game client is hosting a game session and that there are two members (players) in the game session.

8. Close both clients by running the `quit` command in both game consoles.

Congratulations, you have successfully created a multiplayer-enabled game project!

We’d love to hear from you! Head to our Tutorial Discussion forum to share any feedback you have, including what you do or don’t like about our tutorials or new content you’d like to see in the near future.