

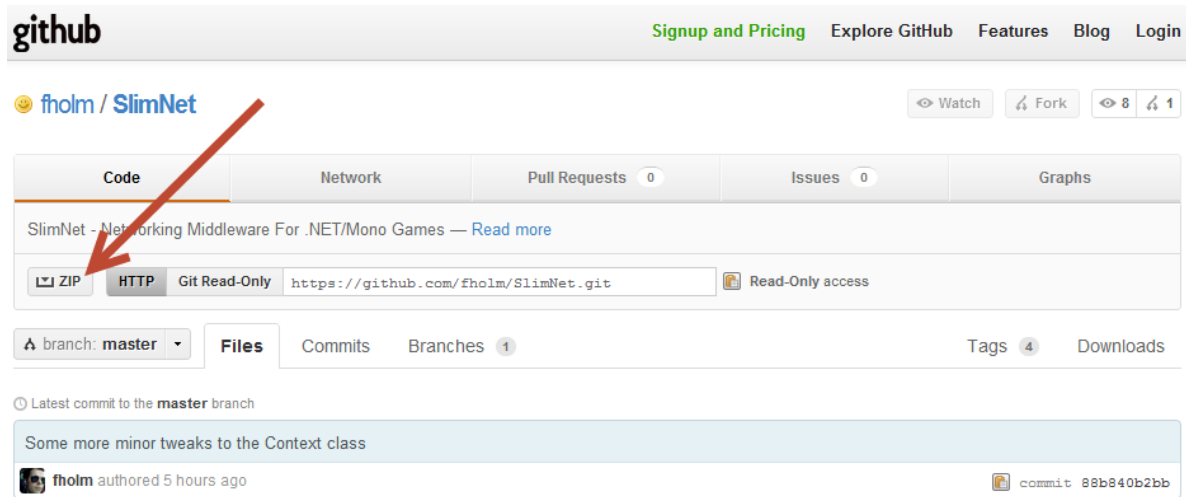
SLIMNET

SETUP INSTRUCTIONS

VERSION 0.2.1

## Getting the source code

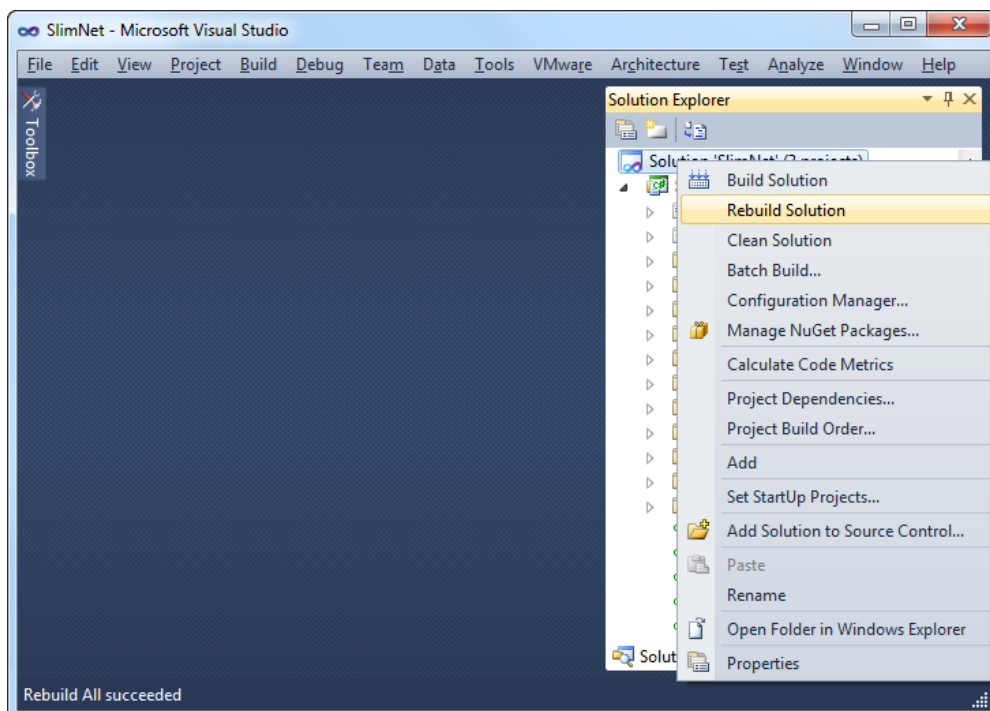
To get the source code for SlimNet you can either download the source code by navigating to <http://github.com/fholm/SlimNet> in your browser and clicking on the ZIP download link and then unpacking it somewhere on your disk.



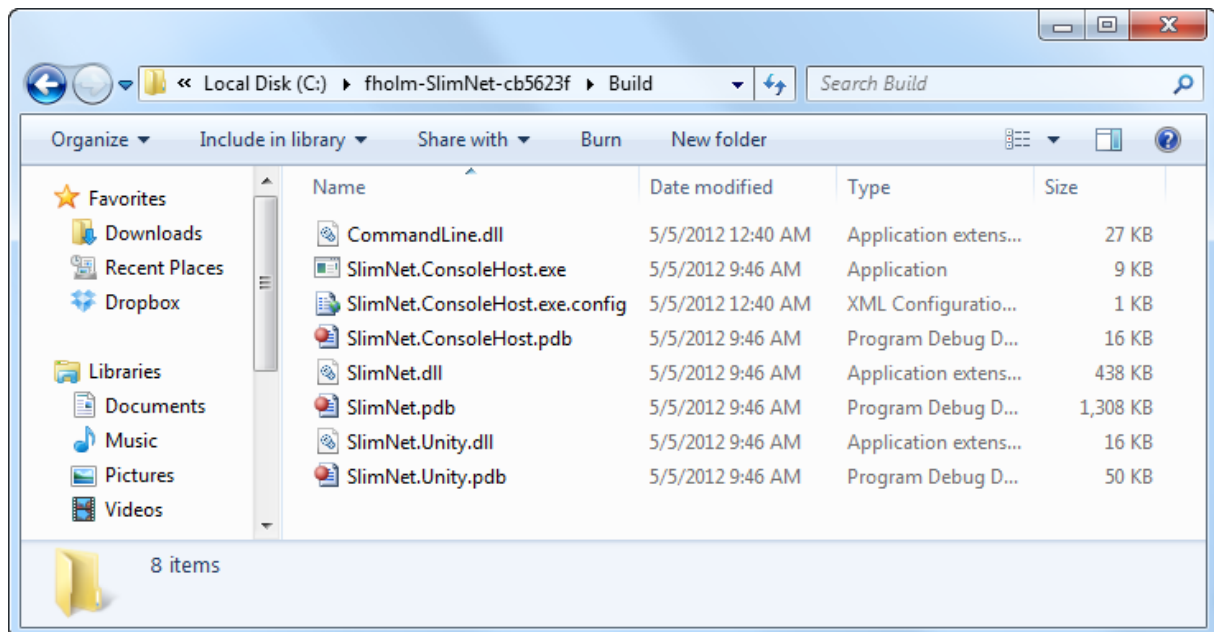
Or you can clone the repository using the Git Read-Only URL: <git://github.com/fholm/SlimNet.git>

## Compiling

The first thing we should do is compile the SlimNet solution itself, these instructions will use Windows and Visual Studio, but it should work on OS X and Mono Develop also. Open the SlimNet/SlimNet.sln file, right click on the solution and select Rebuild Solution.



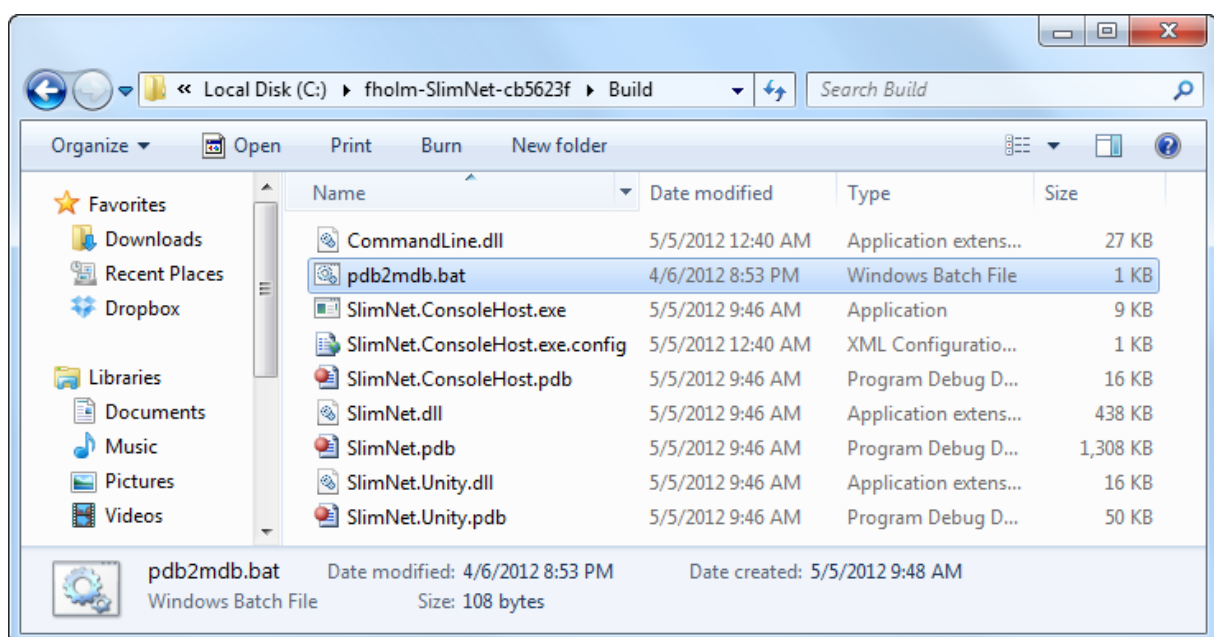
This should create a new folder called **Build** in the main project directory, with the following contents:



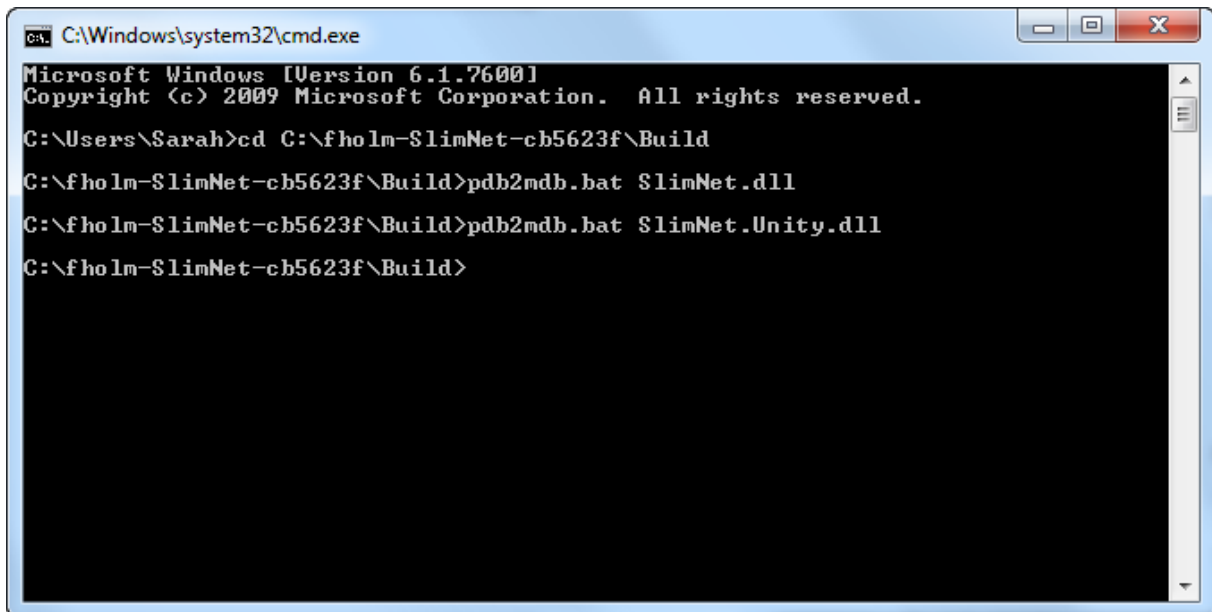
## Generating Mono Debug Symbols

This step is optional, but recommended as it will allow you to debug the SlimNet libraries inside the Unity MonoDevelop debugger. It requires you to have a standalone version of Mono installed; it can be downloaded from <http://www.mono-project.com>.

On windows mono should be located in `C:\Program Files (x86)\Mono-Version` or possibly `C:\Program Files\Mono-Version` depending on if you're on a 64bit or 32bit version of Windows. Navigate to the bin directory inside your mono installation, for me it's located at `C:\Program Files (x86)\Mono-2.10.8\bin`. For some weird reason the pdb2mdb tool we're going to use doesn't like to work on files that aren't in the directory it's being called from, so copy the `pdb2mdb.bat` file from the mono bin directory into the `Build` directory inside the SlimNet project:



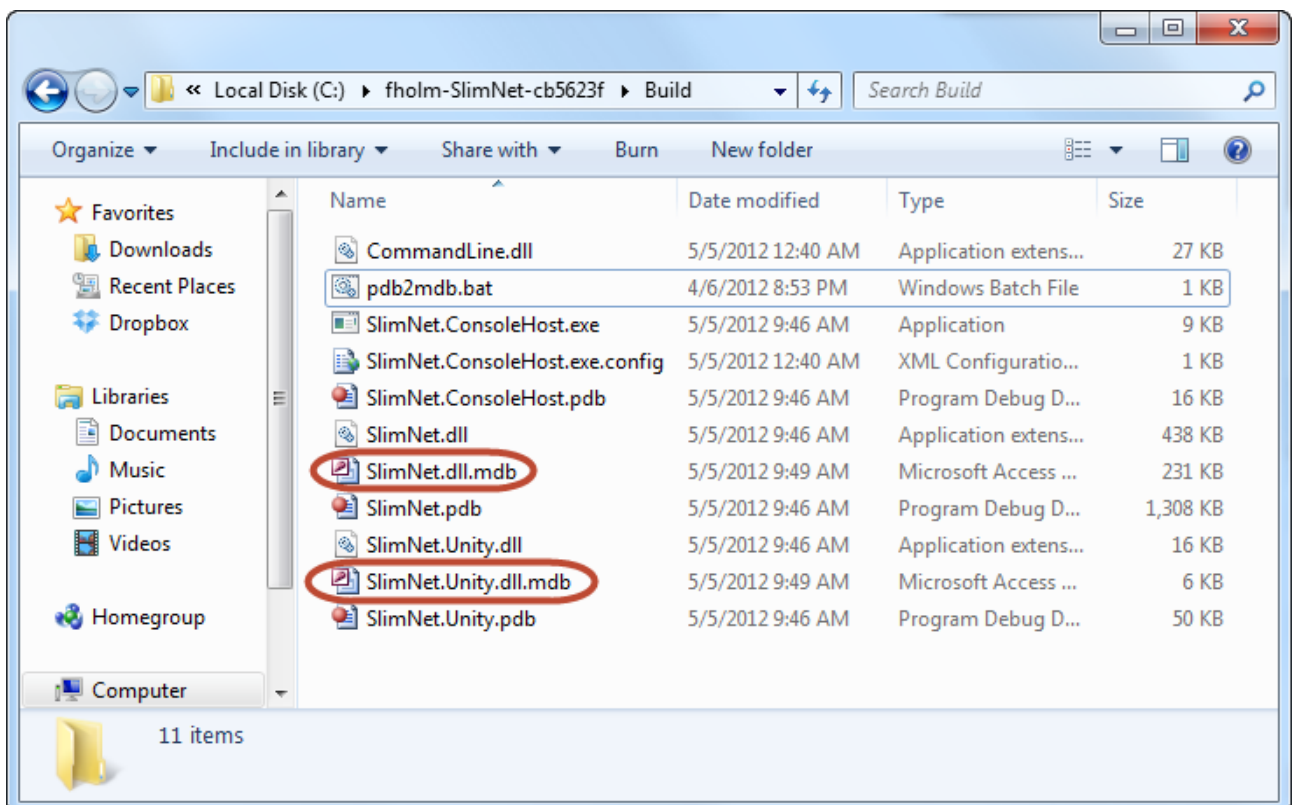
Now open a command prompt and navigate to the Build directory and then run the [pdb2mdb.bat](#) utility on [SlimNet.dll](#) and [SlimNet.Unity.dll](#), like this:



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.












C:\Users\Sarah>cd C:\fholm-SlimNet-cb5623f\Build
C:\fholm-SlimNet-cb5623f\Build>pdb2mdb.bat SlimNet.dll
C:\fholm-SlimNet-cb5623f\Build>pdb2mdb.bat SlimNet.Unity.dll
C:\fholm-SlimNet-cb5623f\Build>
```

This will produce two new files in the Build directory called [SlimNet.dll.mdb](#) and [SlimNet.Unity.dll.mdb](#).

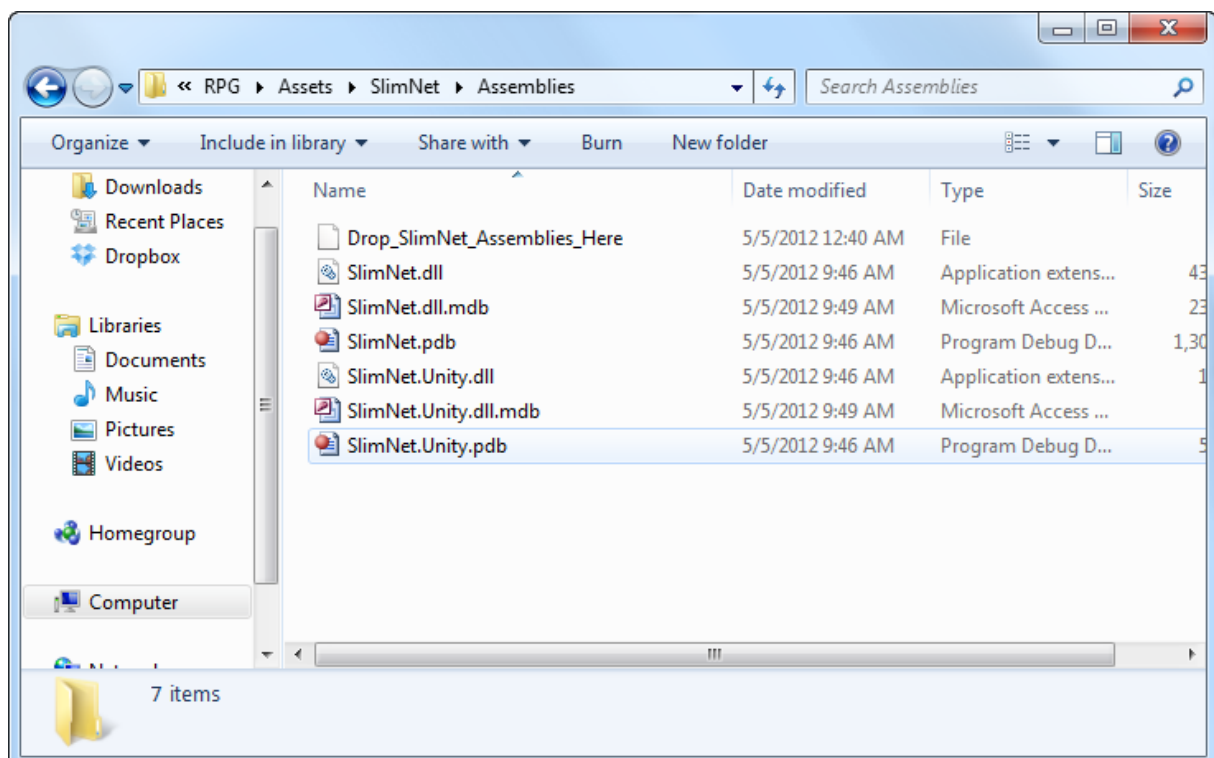


Now select and copy these six files

- SlimNet.dll
- SlimNet.pdb
- SlimNet.dll.mdb
- SlimNet.Unity.dll
- SlimNet.Unity.pdb
- SlimNet.Unity.dll.mdb

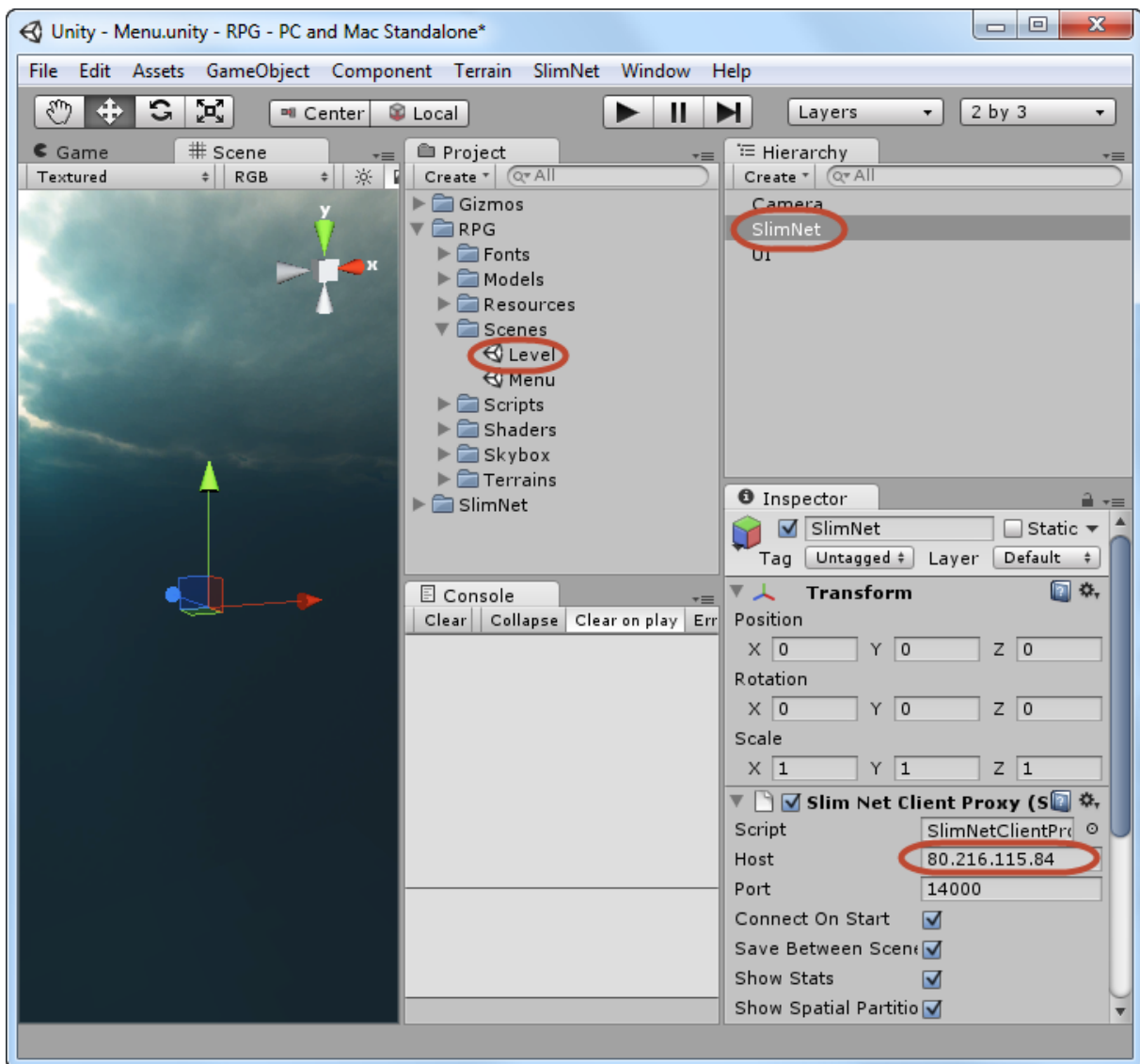
Name	Date modified	Type	Size
 CommandLine.dll	5/5/2012 12:40 AM	Application extens...	27 KB
 pdb2mdb.bat	4/6/2012 8:53 PM	Windows Batch File	1 KB
 SlimNet.ConsoleHost.exe	5/5/2012 9:46 AM	Application	9 KB
 SlimNet.ConsoleHost.exe.config	5/5/2012 12:40 AM	XML Configuratio...	1 KB
 SlimNet.ConsoleHost.pdb	5/5/2012 9:46 AM	Program Debug D...	16 KB
 SlimNet.dll	5/5/2012 9:46 AM	Application extens...	438 KB
 SlimNet.dll.mdb	5/5/2012 9:49 AM	Microsoft Access ...	231 KB
 SlimNet.pdb	5/5/2012 9:46 AM	Program Debug D...	1,308 KB
 SlimNet.Unity.dll	5/5/2012 9:46 AM	Application extens...	16 KB
 SlimNet.Unity.dll.mdb	5/5/2012 9:49 AM	Microsoft Access ...	6 KB
 SlimNet.Unity.pdb	5/5/2012 9:46 AM	Program Debug D...	50 KB

And paste them into the `Demo\RPG\Assets\SlimNet\Assemblies` folder:

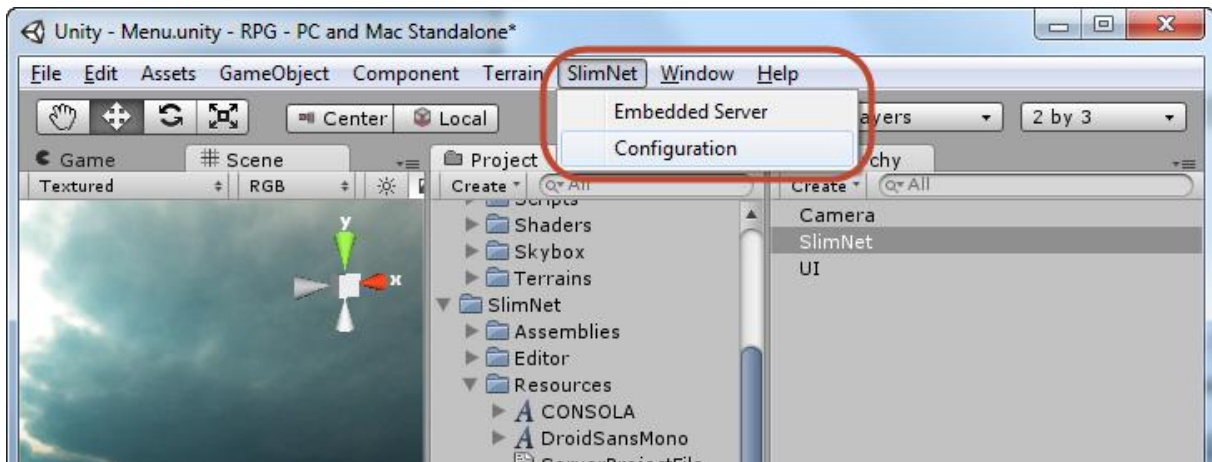


## Configuring Unity

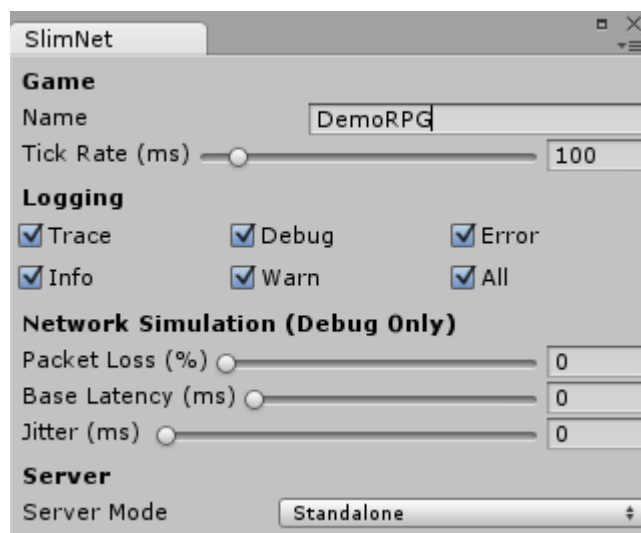
You can now open the project in Unity. Open the RPG/Scenes/Level scene, and select the SlimNet game object in the scene hierarchy. First thing you probably want to do is to change the Host setting to something like 127.0.0.1 if you're going to run the server on localhost or in embedded mode inside the editor while developing (as we will in this guide).



The next thing we're doing is opening the SlimNet configuration tab by selecting SlimNet/Configuration option from the menu:



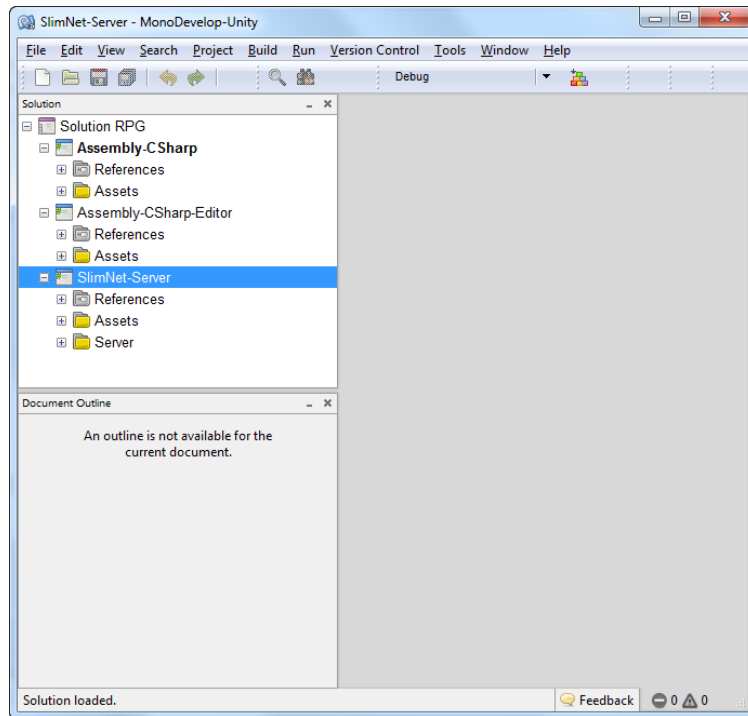
Once you've opened the configuration tab there are a bunch of settings here, the only really important one is the Name one for now, set this to "DemoRPG".



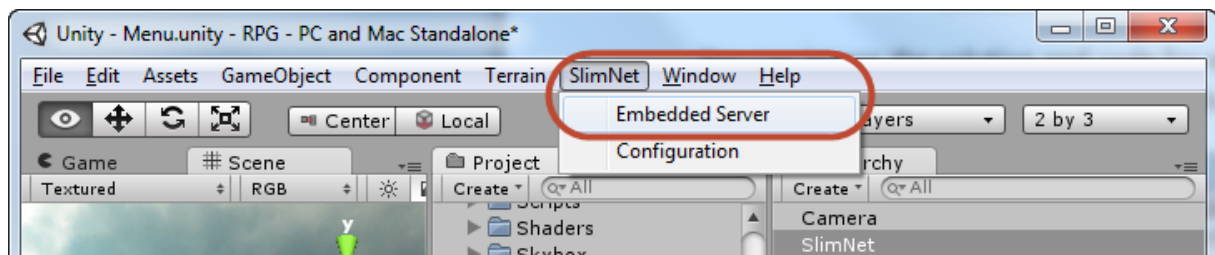
You might also want to turn off the Trace logging option as it creates a lot of spam in the in game console. You can experiment with tick rate; I find that leaving it at 100ms gives adequate results but you might want to change it to 50ms for a smoother experience.

The next thing we're going to do is to select the Assets/Sync MonoDevelop Project from the Unity menu, this will generate the solution file and its associated project files in the [Demo/RPG](#) folder, but it will also tell SlimNet to generate its own [SlimNet-Server.csproj](#) file inside the same folder and inject it into unity solution file.

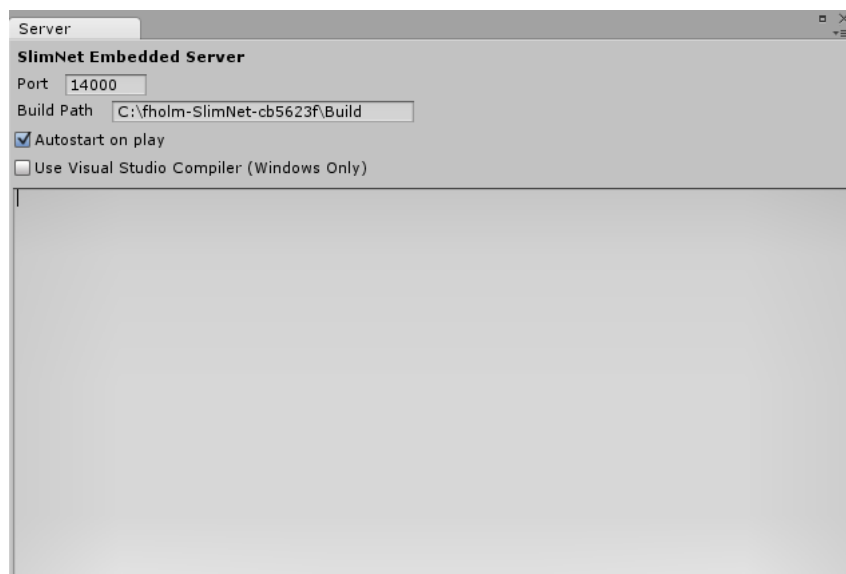
When you open the solution in Visual Studio you will now see three projects, instead of the default two ones that Unity produce, the third one contains the code for the server part of our game.



You can ignore the solution and code for now, instead go back to Unity and select the SlimNet/Embedded Server menu option from the top bar:

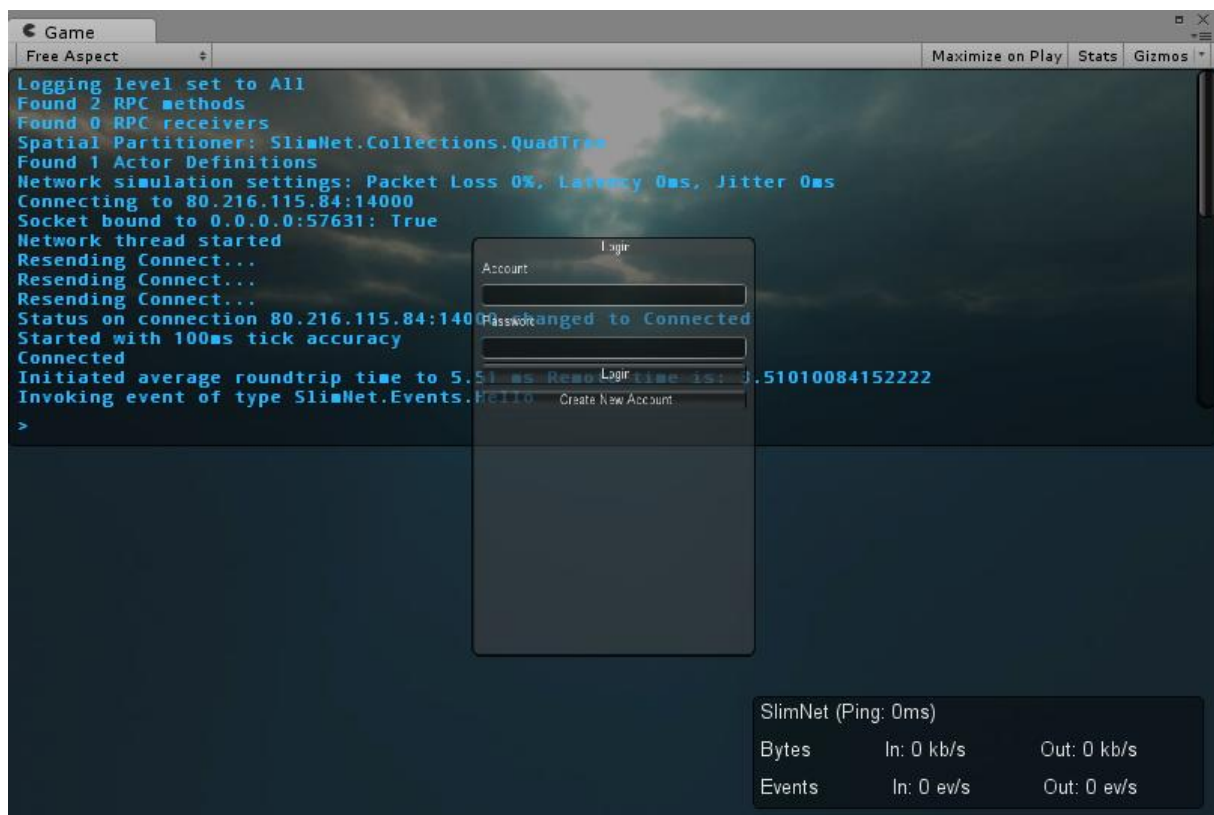
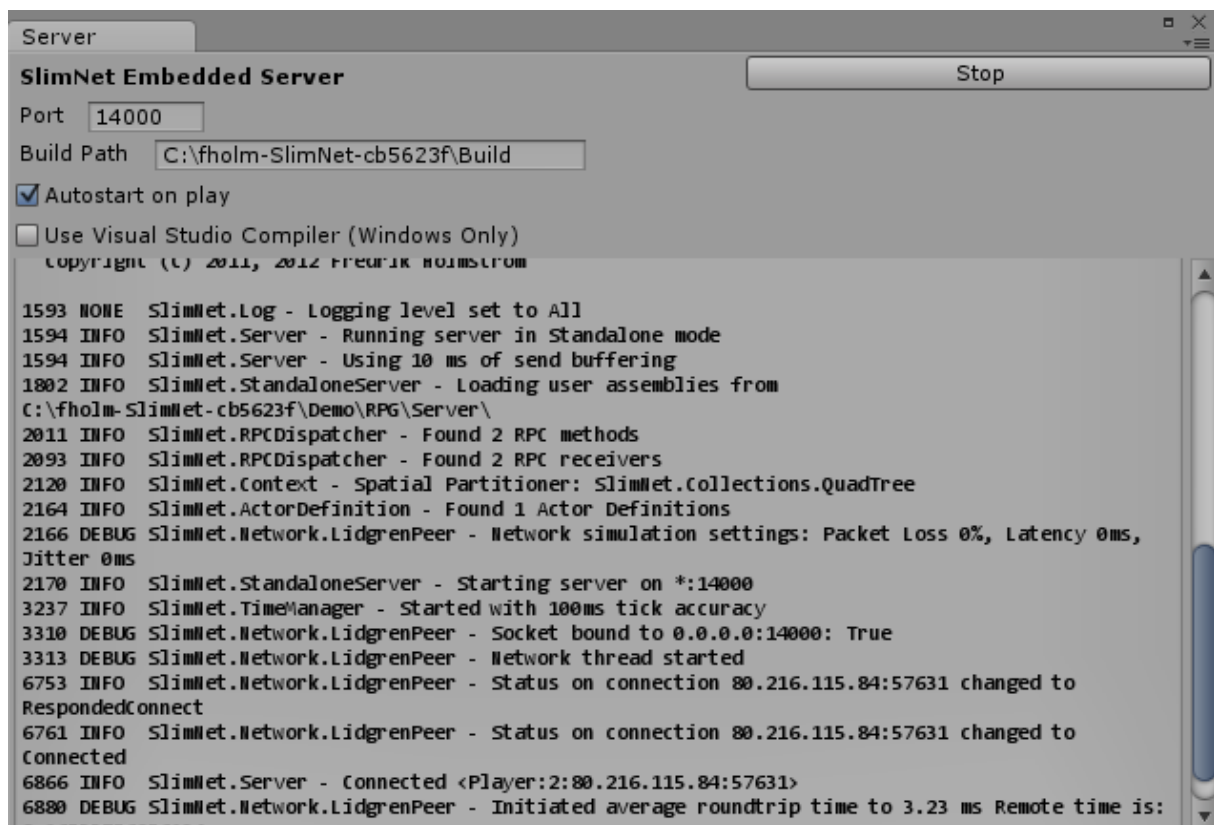


This will open a new tab window in Unity that contains the SlimNet embedded server, check Autostart on play. There are two settings we need to do here, first set the Build Path setting to where we have the Build directory on disk, and then also check Autostart on play.

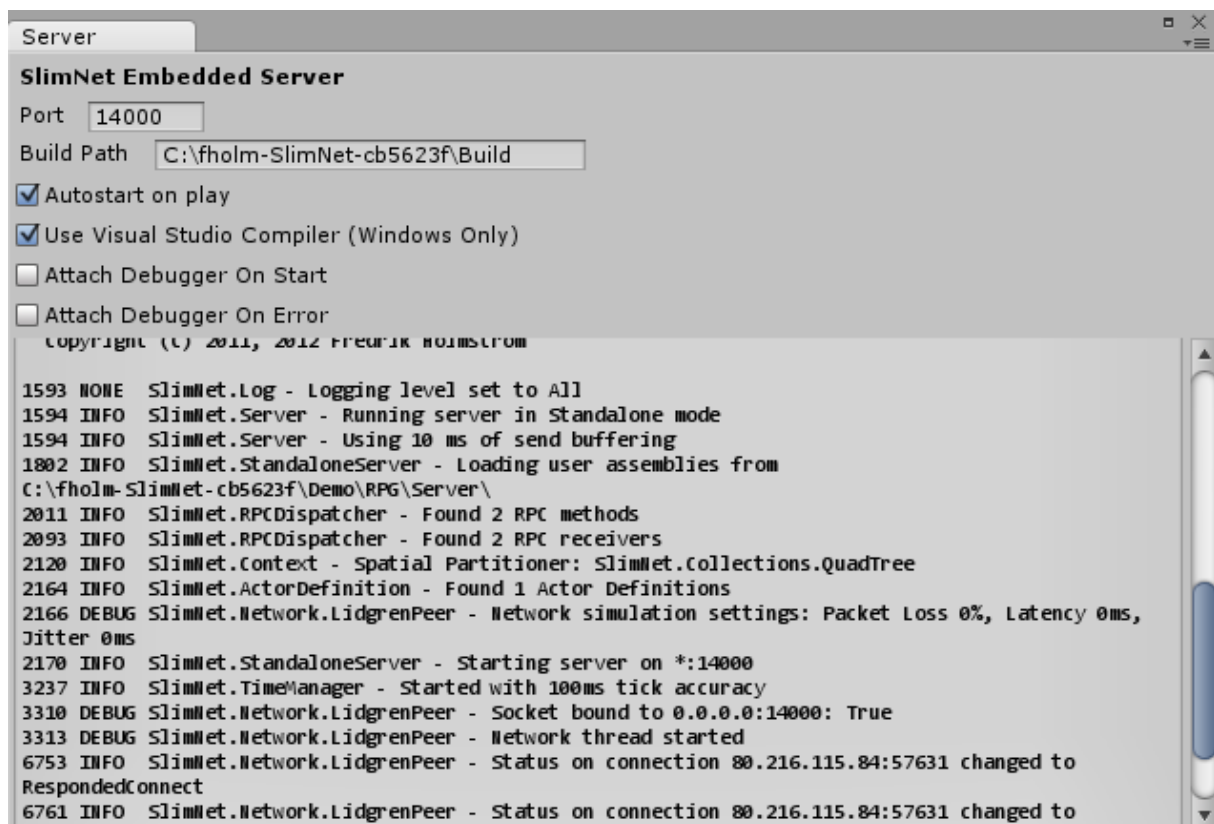




You should now be able to hit Play in the Unity editor, and the server and client will start up:



Now, if you're on windows and you want to debug the server code running in the Embedded Server, you can check the option "Use Visual Studio Compiler" and you will get two other options to attach the debugger on error or attach the debugger directly on start:



This will automatically attach the Visual Studio debugger to your server code. That's all folks, enjoy!