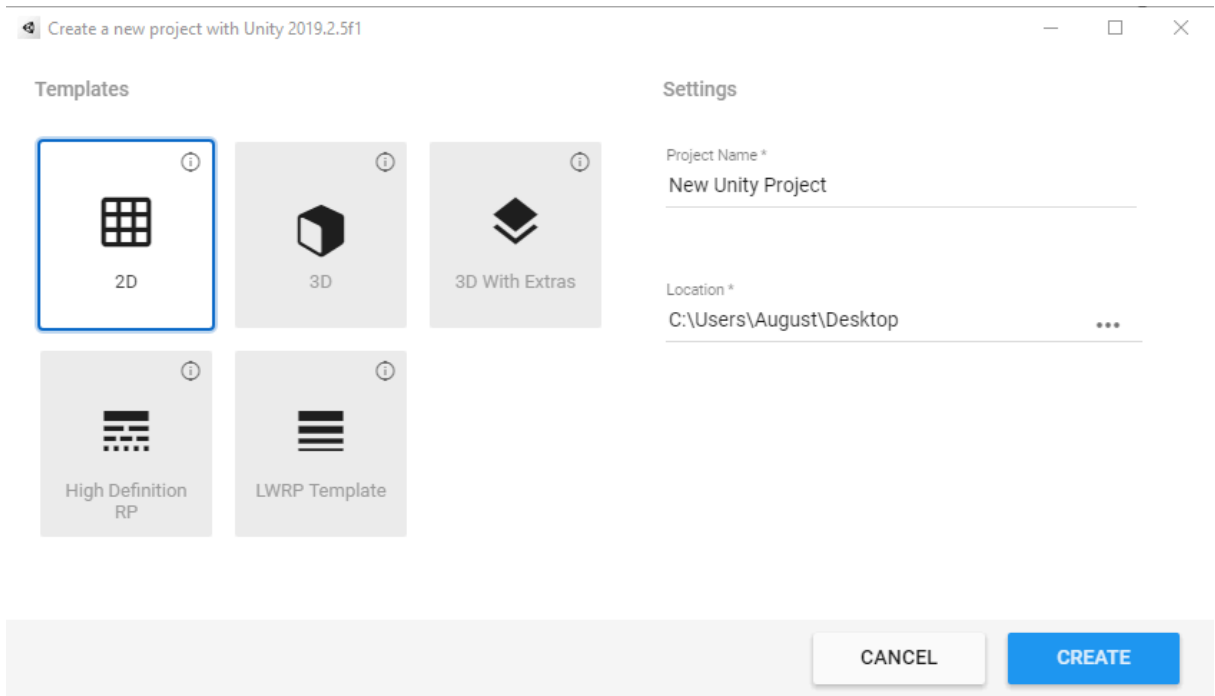
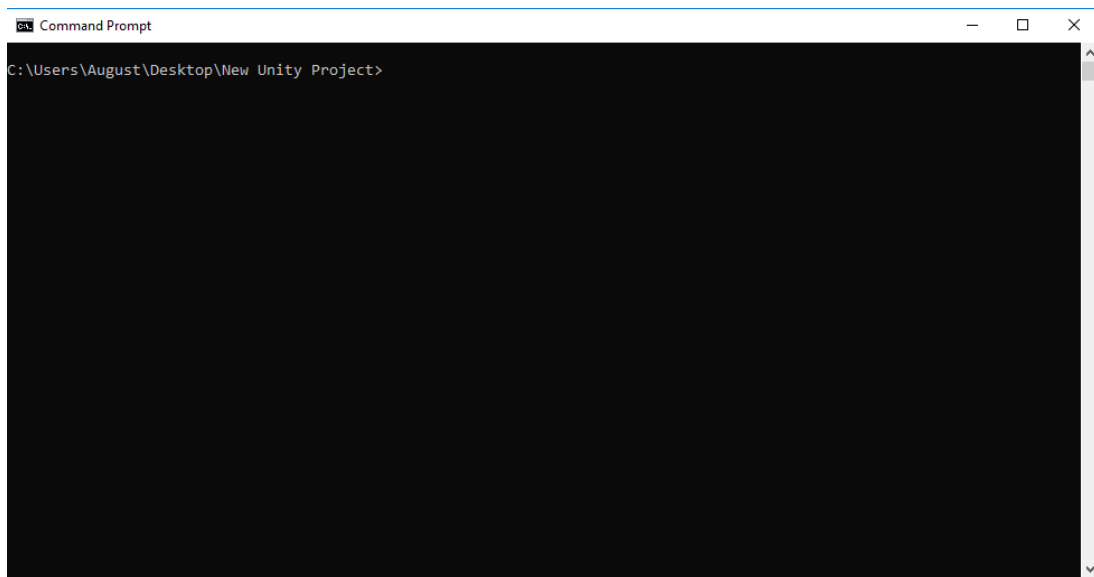


## Setting up F# with a Unity Project (Unity 2019.2.5f1)

1. Install the [.NET Core SDK](#)
2. Create a Unity project. I'll use my Desktop as my home directory.

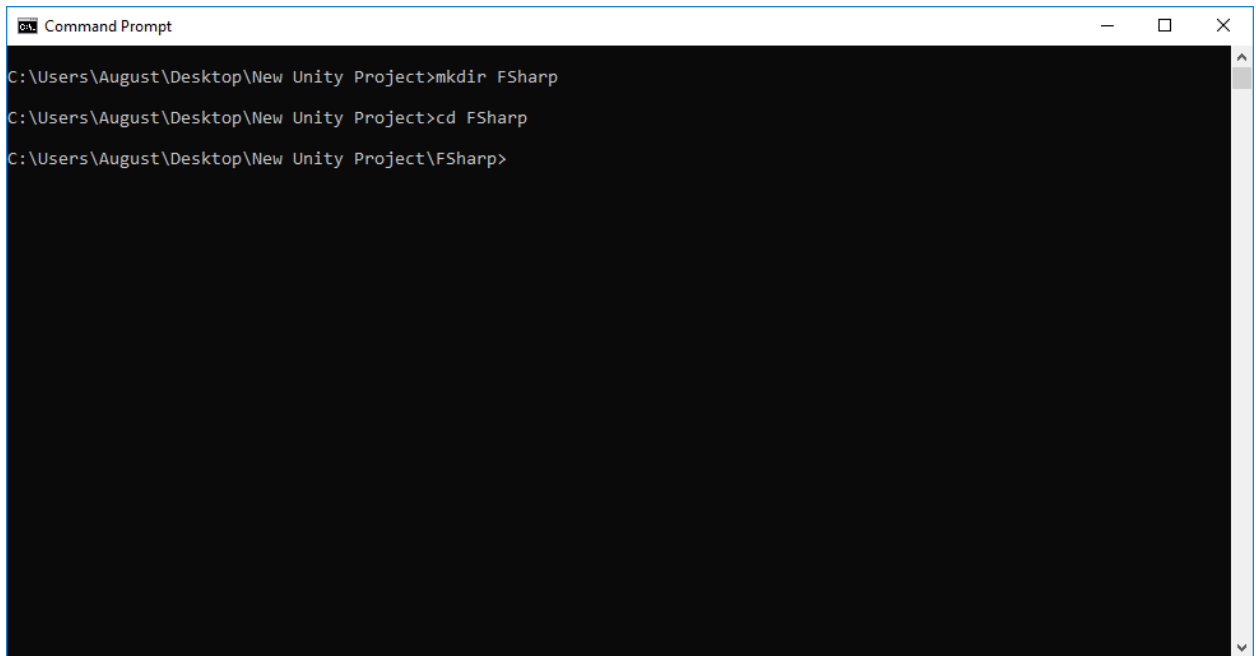


3. Navigate to the Unity Project in a command prompt, or an application that supports executing bash commands.



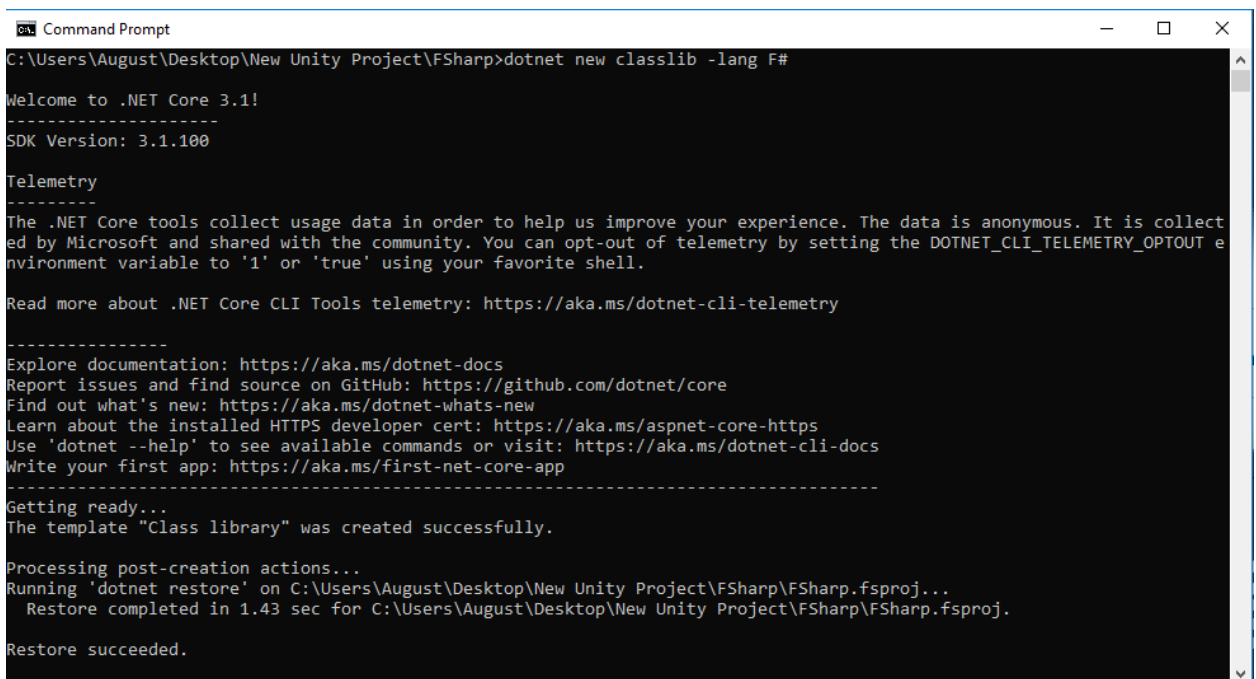
4. Make a new directory using the command: **mkdir FSharp**

5. Enter the new directory using the command: **cd FSharp**



```
Command Prompt
C:\Users\August\Desktop\New Unity Project>mkdir FSharp
C:\Users\August\Desktop\New Unity Project>cd FSharp
C:\Users\August\Desktop\New Unity Project\FSharp>
```

6. Enter this command to create a new FSharp Class library that we will set up so Unity can interact with it: **dotnet new classlib -lang F#**



```
Command Prompt
C:\Users\August\Desktop\New Unity Project\FSharp>dotnet new classlib -lang F#

Welcome to .NET Core 3.1!
-----
SDK Version: 3.1.100

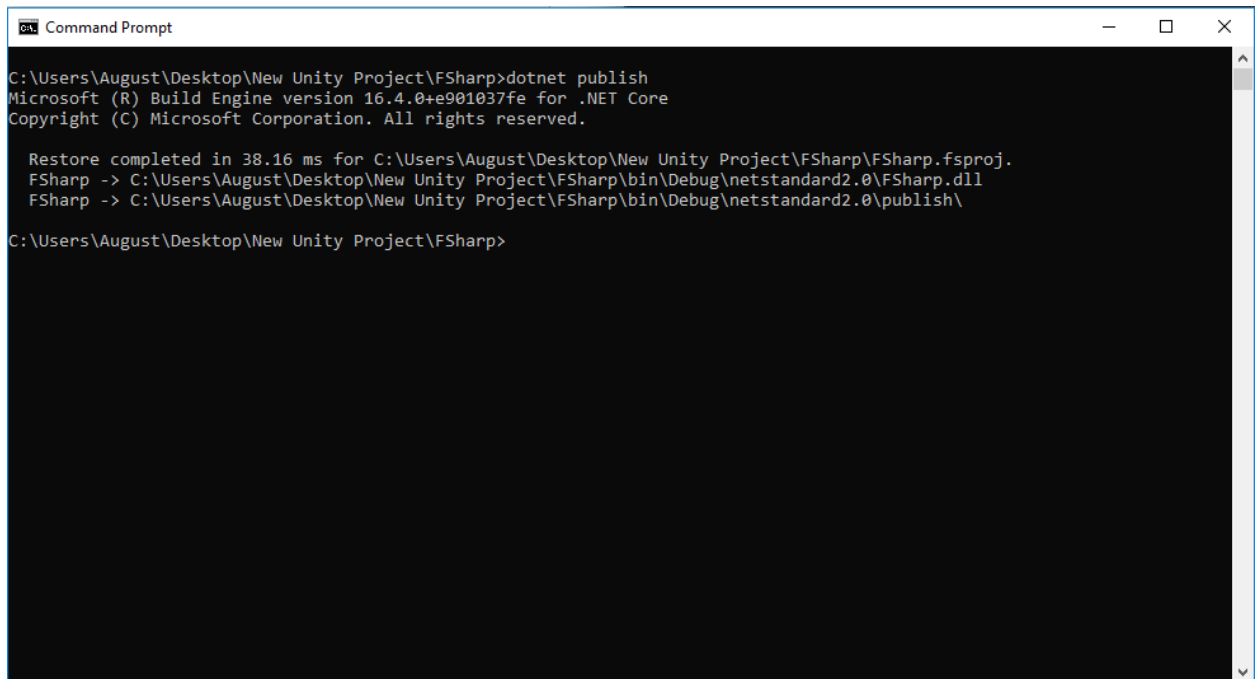
Telemetry
-----
The .NET Core tools collect usage data in order to help us improve your experience. The data is anonymous. It is collected by Microsoft and shared with the community. You can opt-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT environment variable to '1' or 'true' using your favorite shell.

Read more about .NET Core CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

-----
Explore documentation: https://aka.ms/dotnet-docs
Report issues and find source on GitHub: https://github.com/dotnet/core
Find out what's new: https://aka.ms/dotnet-whats-new
Learn about the installed HTTPS developer cert: https://aka.ms/aspnet-core-https
Use 'dotnet --help' to see available commands or visit: https://aka.ms/dotnet-cli-docs
Write your first app: https://aka.ms/first-net-core-app
-----
Getting ready...
The template "Class library" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on C:\Users\August\Desktop\New Unity Project\FSharp\FSharp.fsproj...
  Restore completed in 1.43 sec for C:\Users\August\Desktop\New Unity Project\FSharp\FSharp.fsproj.
Restore succeeded.
```

7. Enter this command to build the FSharp project dll: **dotnet publish**

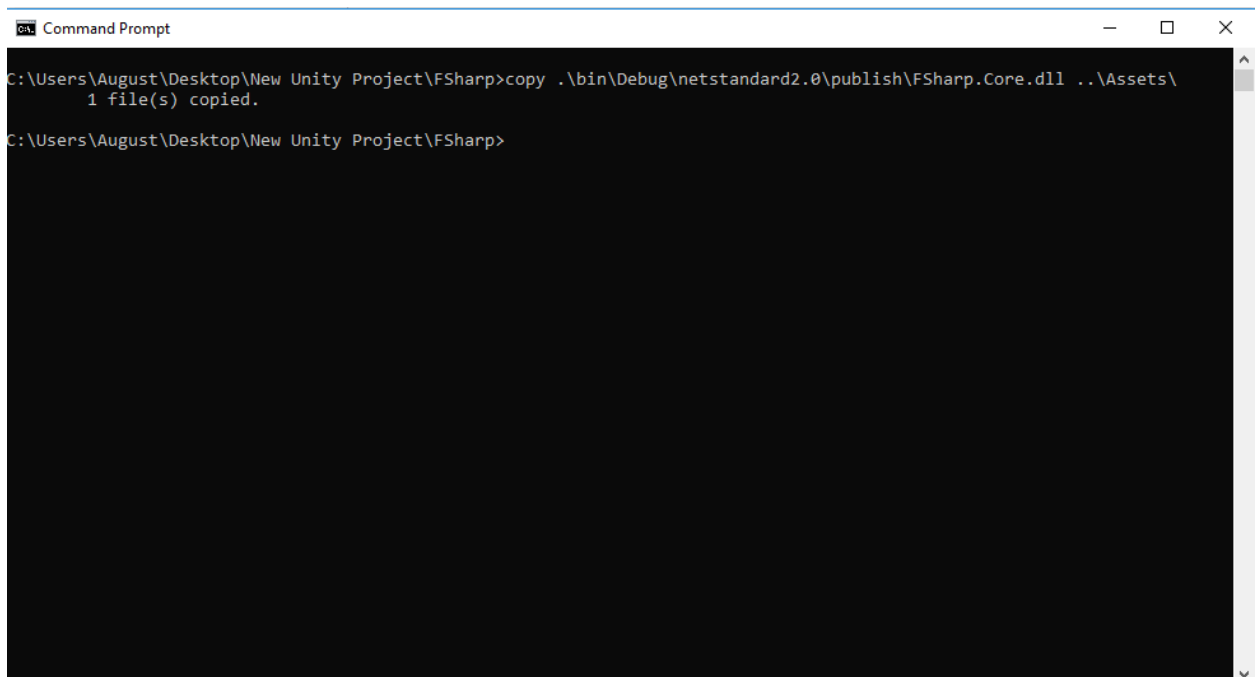


```
Command Prompt

C:\Users\August\Desktop\New Unity Project\FSharp>dotnet publish
Microsoft (R) Build Engine version 16.4.0+e901037fe for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

Restore completed in 38.16 ms for C:\Users\August\Desktop\New Unity Project\FSharp\FSharp.fsproj.
FSharp -> C:\Users\August\Desktop\New Unity Project\FSharp\bin\Debug\netstandard2.0\FSharp.dll
FSharp -> C:\Users\August\Desktop\New Unity Project\FSharp\bin\Debug\netstandard2.0\publish\
C:\Users\August\Desktop\New Unity Project\FSharp>
```

8. Enter this command to move the dll (holding all the class library files in our F# project) into the Unity Project Assets folder: **copy**  
**.\bin\Debug\netstandard2.0\publish\FSharp.Core.dll ..\Assets\**

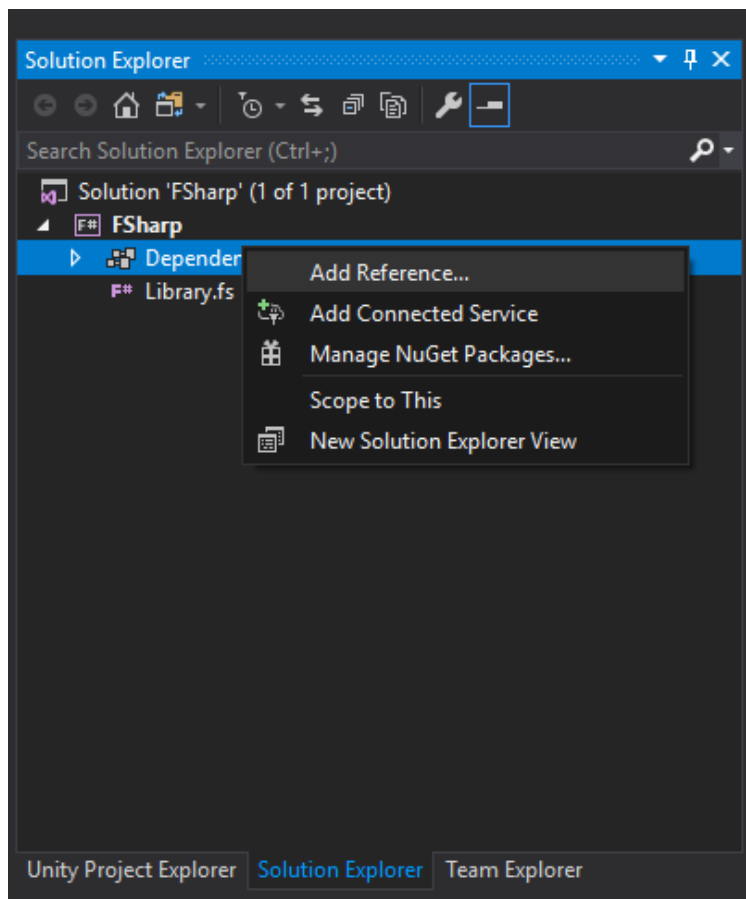


```
Command Prompt

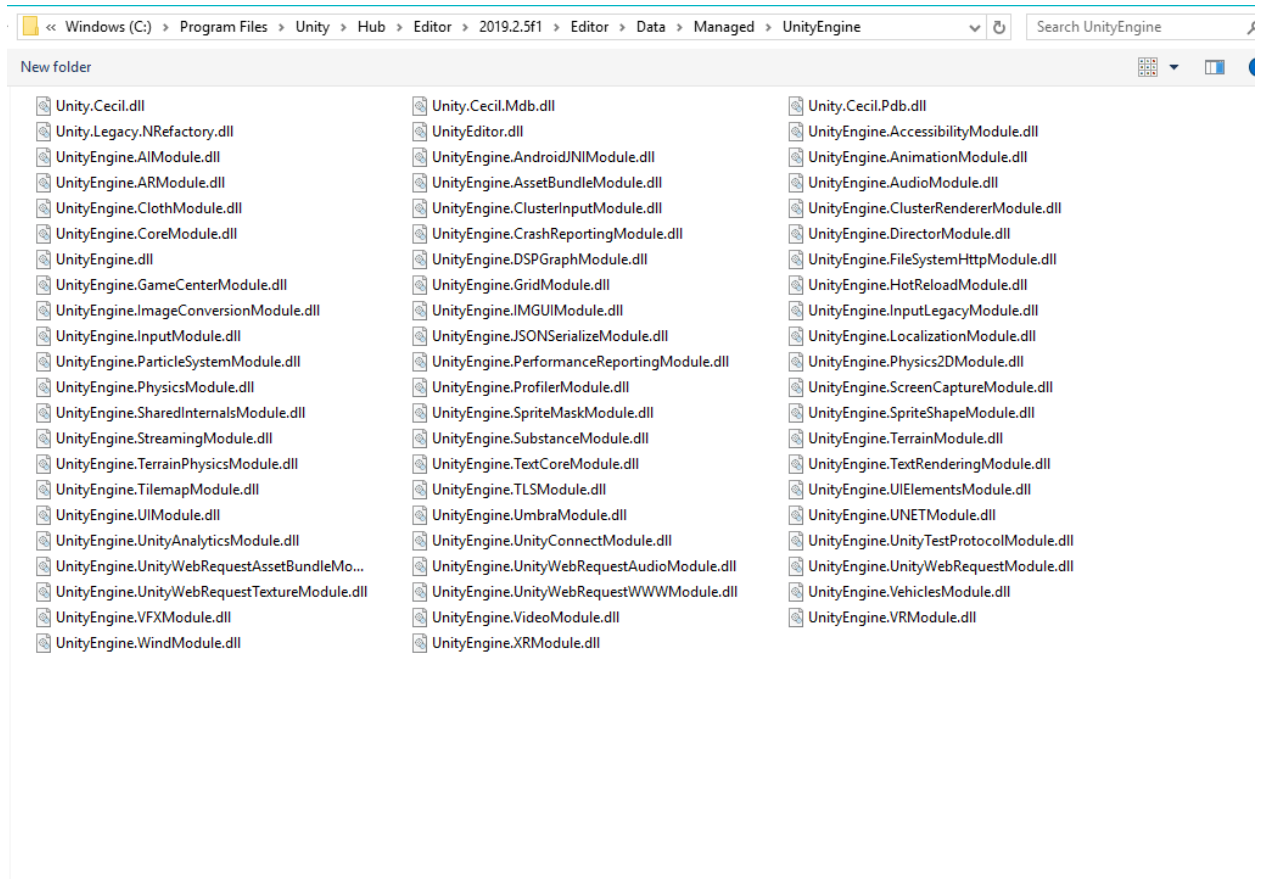
C:\Users\August\Desktop\New Unity Project\FSharp>copy .\bin\Debug\netstandard2.0\publish\FSharp.Core.dll ..\Assets\
1 file(s) copied.
C:\Users\August\Desktop\New Unity Project\FSharp>
```

9. Open the .FSPROJ File in visual studio

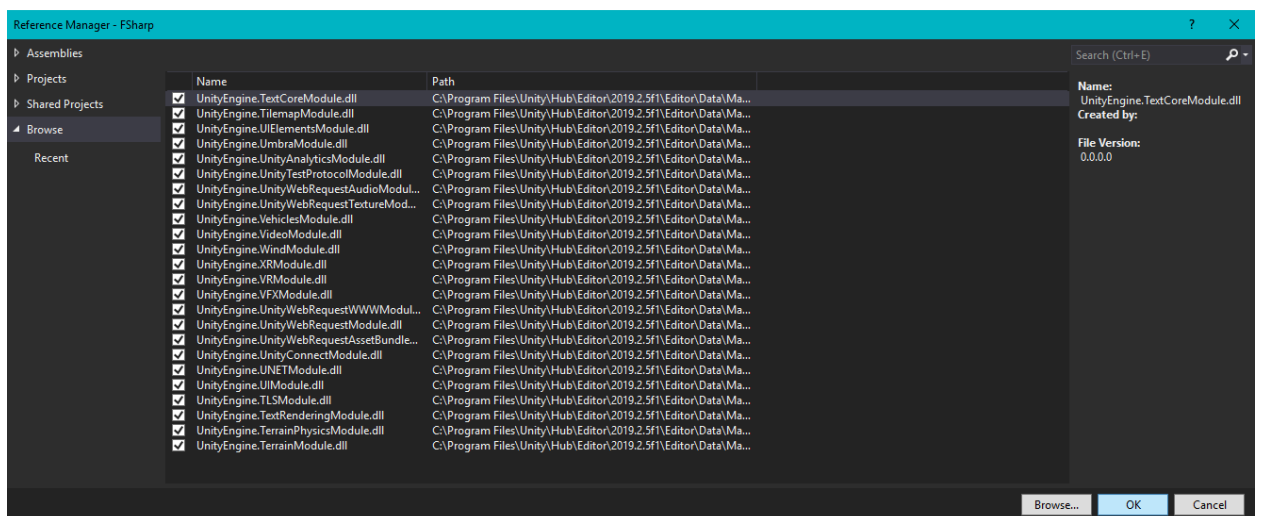
10. In the Solution Explorer window, expand the FSharp Solution and right-click on Dependencies. From there, click Add Reference...



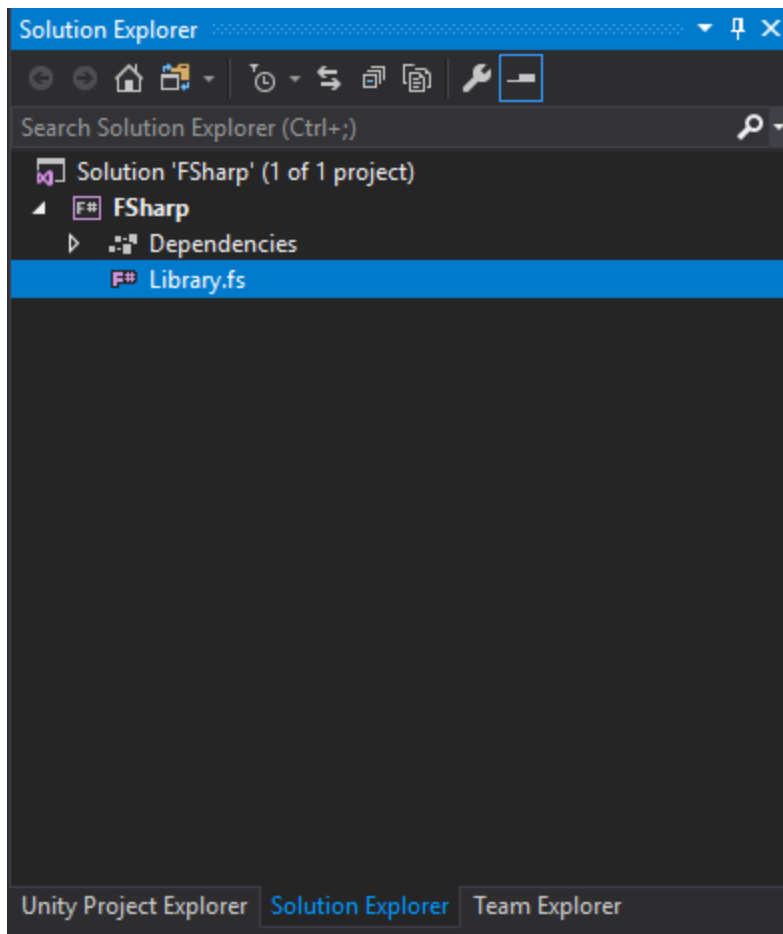
11. Click the Browse... button and enter the path to your Unity install directory and select the directory ending in /Contents/Managed/UnityEngine



12. Select all the dll's and then click OK to finish adding the references to the project



13. Open up Library.fs by double-clicking on it in the Solution Explorer



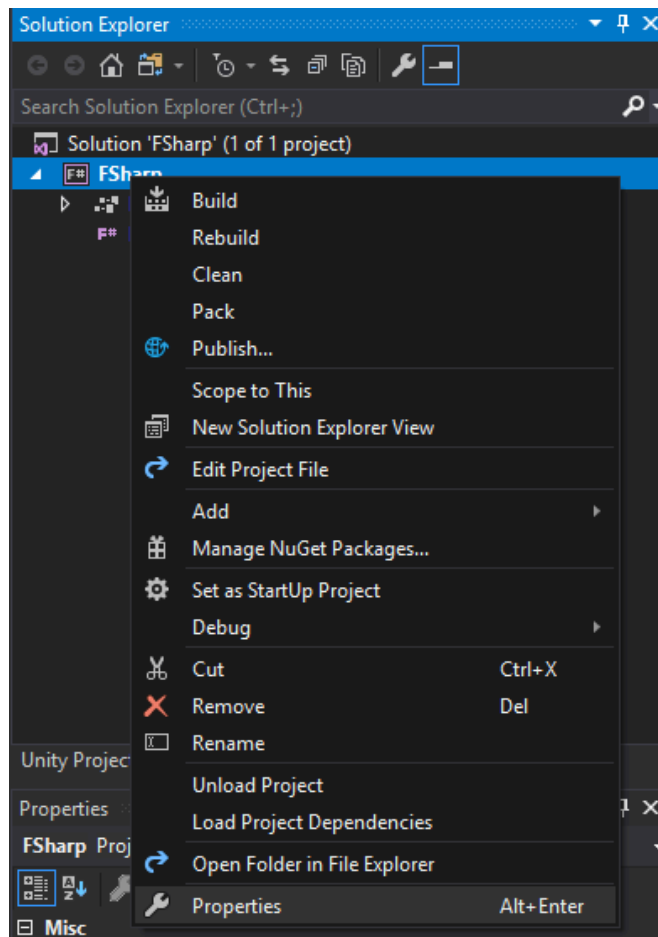
14. Modify the code inside Library.fs to look like this:

```
namespace FSharp

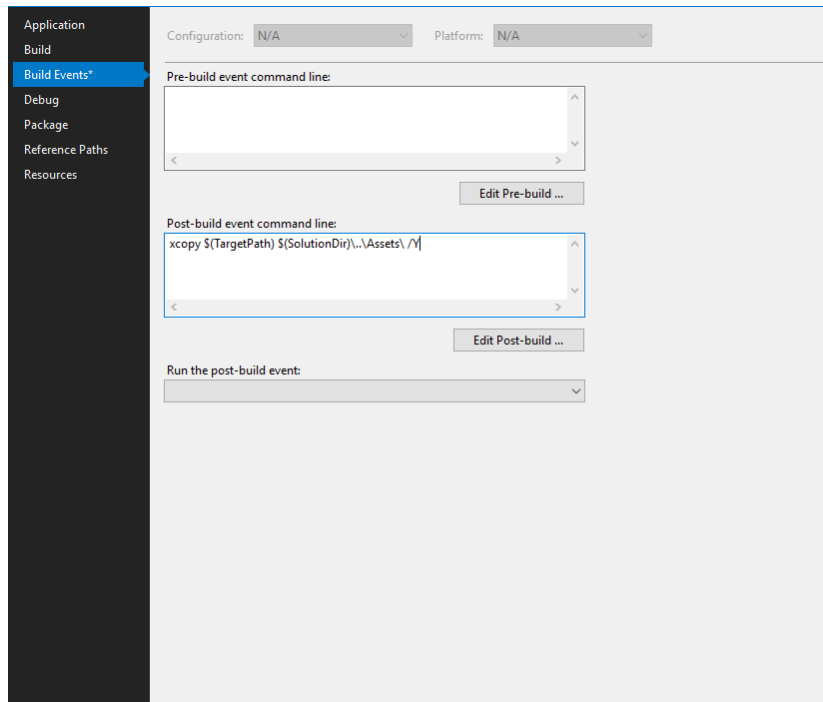
open UnityEngine

type SimpleScript() =
    inherit MonoBehaviour()
    member this.Start() = Debug.Log("Hello World!")
```

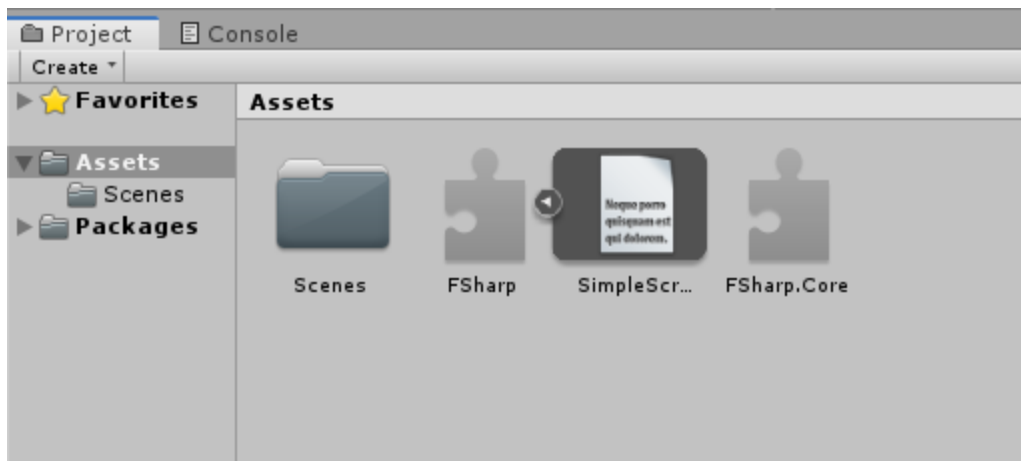
15. Right-click on the FSharp Project in the Solution Explorer and click Properties



16. Add the post-build event command: **xcopy \$(TargetPath) \$(SolutionDir)\..\Assets\ /Y**

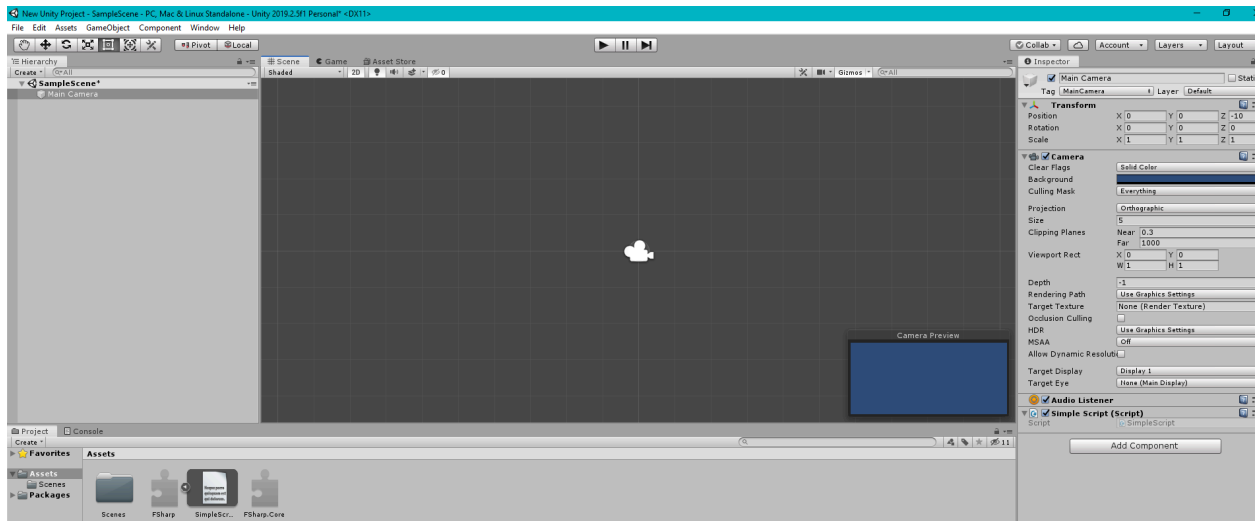


17. Build the project (CTRL+SHIFT+B)
18. Go back to the Unity Editor and see that your project's DLL has been copied over to your Asset folder. Also notice that within the FSharp DLL, SimpleScript is present.

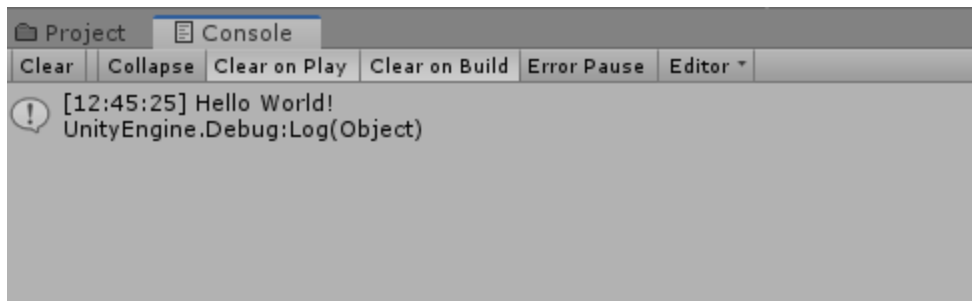




19. Click and drag SimpleScript onto the camera in the game scene



20. Run the game. In the console, you should be able to see: **Hello World!** printed out.



This is the conclusion of the tutorial. Your F# project scripts can now be used in Unity!