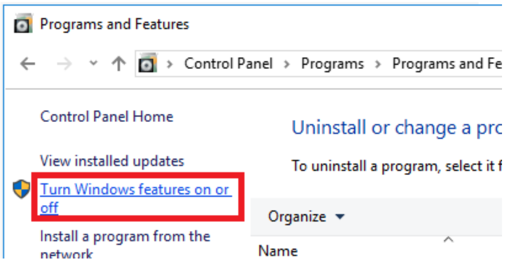
For unity2017:

To be able to compile .net libraries for unity, you need to use .net framework 3.5, or less. Since this is legacy .net, you need to configure Windows in a certain way.

1. Make sure you have .net 3.5 installed.

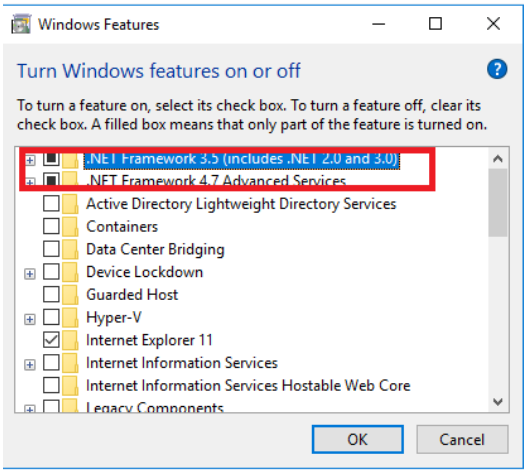
This is not available directly from the visual studio installer, you will need to do this:

a) Open your control panel. Click Uninstall a program in Programs. Choose Turn Windows features on or off.


Programs and Features
Control Panel › Programs › Programs and Fe
Control Panel Home
View installed updates
Turn Windows features on or off
Install a program from the network
Uninstall or change a prc
Organize
Name

To uninstall a program, select it f

b) Make sure .net 3.5 is well checked

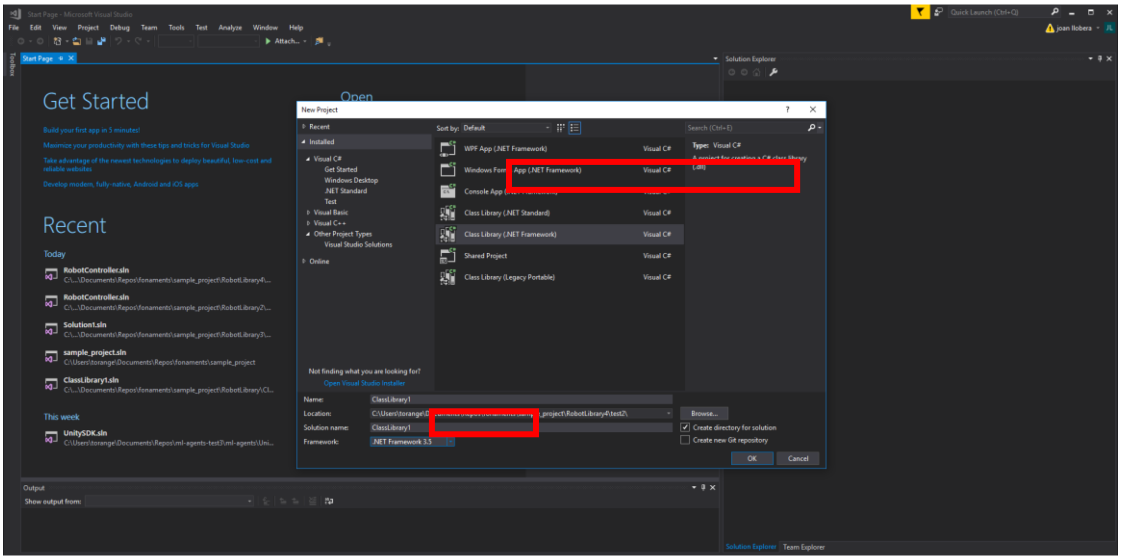


To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ■ .NET Framework 3.5 (includes .NET 2.0 and 3.0)
- ■ .NET Framework 4.7 Advanced Services
- □ Active Directory Lightweight Directory Services
- □ Containers
- □ Data Center Bridging
- □ Device Lockdown
- □ Guarded Host
- □ Hyper-V
- ☑ Internet Explorer 11
- □ Internet Information Services
- □ Internet Information Services Hostable Web Core
- □ Legacy Components

c) Restart the computer

2. Create a new project using the .net 3.5 framework
3. Drag and drop the resulting .dll into the Unity asset folders and it should work out of the box.



RobotController.sln
C:\...\Documents\Repos\fonaments\sample_project\RobotLibrary4\...

RobotController.sln
C:\...\Documents\Repos\fonaments\sample_project\RobotLibrary2\...

Solution1.sln
C:\...\Documents\Repos\fonaments\sample_project\RobotLibrary3\...

sample_project.sln
C:\Users\torange\Documents\Repos\fonaments\sample_project

ClassLibrary1.sln
C:\...\Documents\Repos\fonaments\sample_project\RobotLibrary\Cl...

UnitySDK.sln
C:\Users\torange\Documents\Repos\ml-agents-test3\ml-agents\Uni...

WPF App (.NET Framework)	Visual C#
Windows Form App (.NET Framework)	Visual C#
Console App (.NET Framework)	Visual C#
Class Library (.NET Standard)	Visual C#
Class Library (.NET Framework)	Visual C#
Shared Project	Visual C#
Class Library (Legacy Portable)	Visual C#

Type: Visual C#
A project for creating a C# class library (.dll)

Name: ClassLibrary1
Location: C:\Users\torange\D..._project\RobotLibrary4\test2\
Solution name: ClassLibrary1
Framework: .NET Framework 3.5

☑ Create directory for solution
□ Create new Git repository

References:https://social.msdn.microsoft.com/Forums/es-ES/64b19bac-b5e2-40ce-aea2-efdfd53b1768/install-net-framework-35-for-visual-studio-c-class?forum=netfxbcl https://docs.unity3d.com/Manual/UsingDLL.html https://docs.microsoft.com/en-us/dotnet/core/tutorials/library-with-visual-studio

For unity2019:

The previous is solved, due to the fact that in 2019 versions unity targets .net 4.0, instead of .net3.5

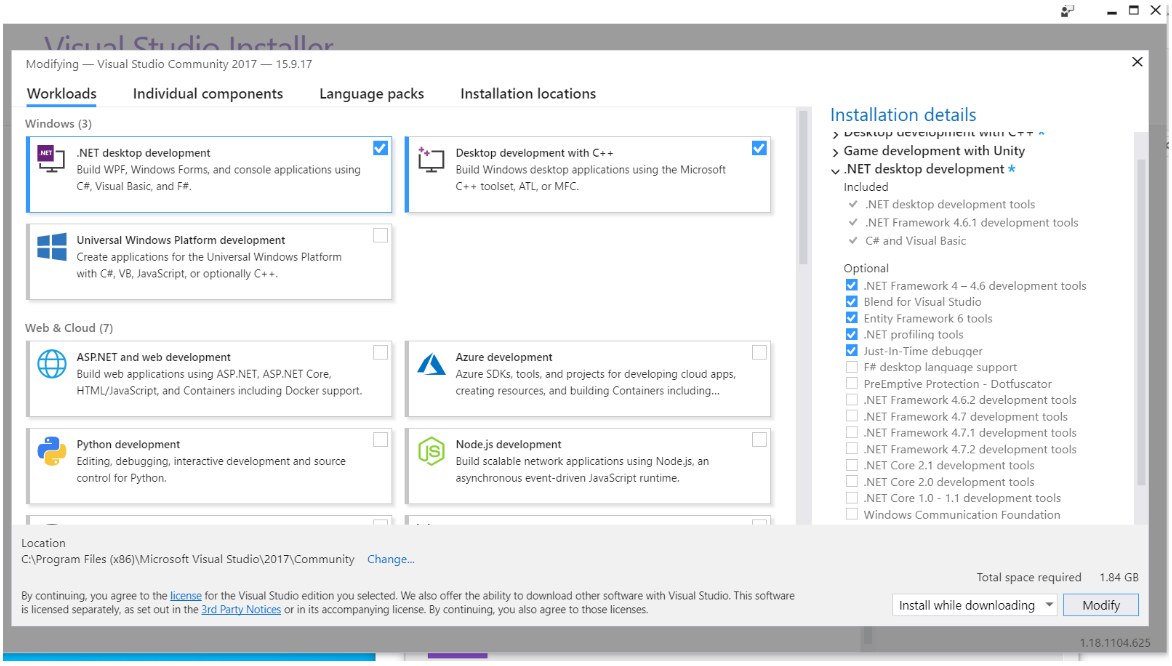
However, in cases where you want to use **UnityEngine.dll** inside your library, you may find the following error:

Severity	Code	Description	Project	File	Line	Suppression State
Warning		The primary reference "UnityEngine" could not be resolved because it has an indirect dependency on the .NET Framework assembly "mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" which has a higher version "4.0.0.0" than the version "2.0.0.0" in the current target framework.	OctopusController			

To address this (reference here).

Please do the following to fix your issue:

1. Quit VS 2017
2. Launch VS 2017 Installer (update it and update visual studio, if necessary)
3. Click Modify.
4. Check the ".NET desktop development" workload.



Modifying — Visual Studio Community 2017 — 15.9.17

Workloads | Individual components | Language packs | Installation locations

Windows (3)

☑ .NET desktop development
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F#.

☑ Desktop development with C++
Build Windows desktop applications using the Microsoft C++ toolset, ATL, or MFC.

□ Universal Windows Platform development
Create applications for the Universal Windows Platform with C#, VB, JavaScript, or optionally C++.

Web & Cloud (7)

□ ASP.NET and web development
Build web applications using ASP.NET, ASP.NET Core, HTML/JavaScript, and Containers including Docker support.

□ Azure development
Azure SDKs, tools, and projects for developing cloud apps, creating resources, and building Containers including...

□ Python development
Editing, debugging, interactive development and source control for Python.

□ Node.js development
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Installation details

› Desktop development with C++ *
› Game development with Unity
⌄ .NET desktop development *

Included
✓ .NET desktop development tools
✓ .NET Framework 4.6.1 development tools
✓ C# and Visual Basic

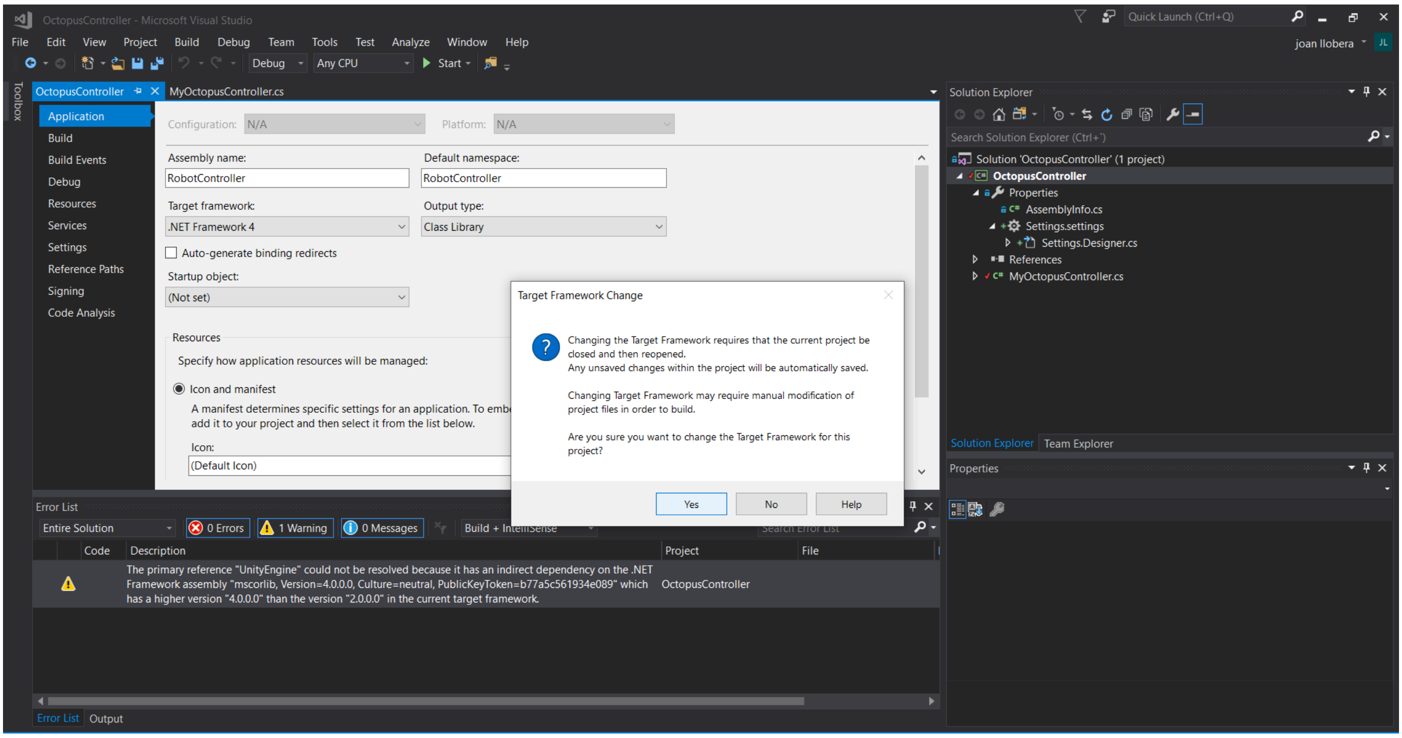
Optional
☑ .NET Framework 4 – 4.6 development tools
☑ Blend for Visual Studio
☑ Entity Framework 6 tools
☑ .NET profiling tools
☑ Just-In-Time debugger
□ F# desktop language support
□ PreEmptive Protection - Dotfuscator
□ .NET Framework 4.6.2 development tools
□ .NET Framework 4.7 development tools
□ .NET Framework 4.7.1 development tools
□ .NET Framework 4.7.2 development tools
□ .NET Core 2.1 development tools
□ .NET Core 2.0 development tools
□ .NET Core 1.0 - 1.1 development tools
□ Windows Communication Foundation

Location
C:\Program Files (x86)\Microsoft Visual Studio\2017\Community Change...

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 1.84 GB

5. Click Modify.
6. Reload your project
7. Right click on the project, open the properties panel, and change the .net target version to 4.0 (See below)



Assembly name: RobotController

Default namespace: RobotController

Target framework: .NET Framework 4

Output type: Class Library

☐ Auto-generate binding redirects

Startup object: (Not set)

Resources

Specify how application resources will be managed:

● Icon and manifest

A manifest determines specific settings for an application. To emb
add it to your project and then select it from the list below.

Icon: (Default Icon)

Target Framework Change

Changing the Target Framework requires that the current project be closed and then reopened.
Any unsaved changes within the project will be automatically saved.

Changing Target Framework may require manual modification of project files in order to build.

Are you sure you want to change the Target Framework for this project?

Yes | No | Help

Error List: Entire Solution | 0 Errors | 1 Warning | 0 Messages | Build + IntelliSense

Code	Description	Project	File
	The primary reference "UnityEngine" could not be resolved because it has an indirect dependency on the .NET Framework assembly "mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" which has a higher version "4.0.0.0" than the version "2.0.0.0" in the current target framework.	OctopusController	

8. Close and reload your project
9. Confirm whether it compiles

Once you are done, you will still have to add a dependency. The simplest way to do this is described here:

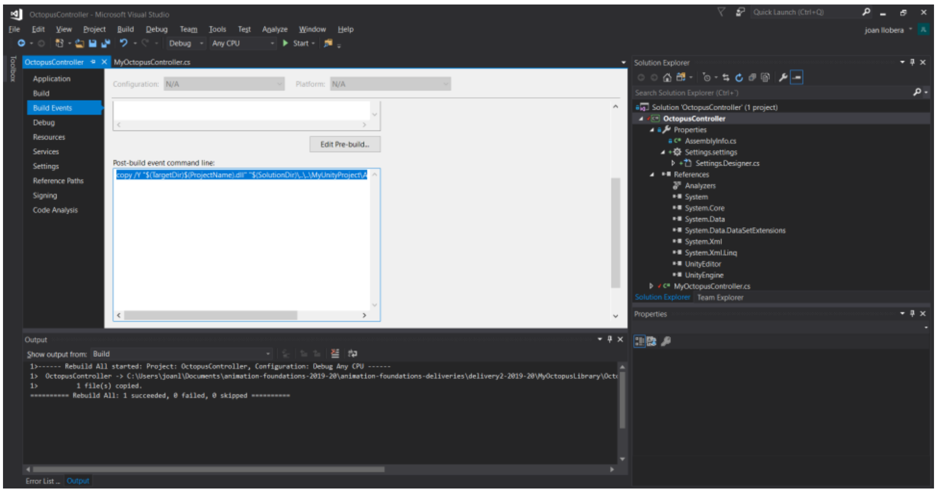
https://docs.unity3d.com/2018.4/Documentation/Manual/UsingDLL.html

For encapsulation purposes, in this project we have chosen to simply add them in a folder, in order the project is self-contained across dependencies

In addition, to automate the copy of the .dll generated inside the unity project, you can add this instruction:

```
copy /Y "$(TargetDir)$(ProjectName).dll" "$(SolutionDir)\..\..\MyUnityProject\Assets\$(ProjectName).dll"
```

This should be added to the post-build events, inside the project properties (see screenshot below).



Post-build event command line:

```
copy /Y "$(TargetDir)$(ProjectName).dll" "$(SolutionDir)\..\..\MyUnityProject\A
```

Output — Show output from: Build

```
1>------ Rebuild All started: Project: OctopusController, Configuration: Debug Any CPU ------
1>  OctopusController -> C:\Users\joanl\Documents\animation-foundations-2019-20\animation-foundations-deliveries\delivery2-2019-20\MyOctopusLibrary\Octo
1>          1 file(s) copied.
========== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==========
```

Debugging in unity2019:

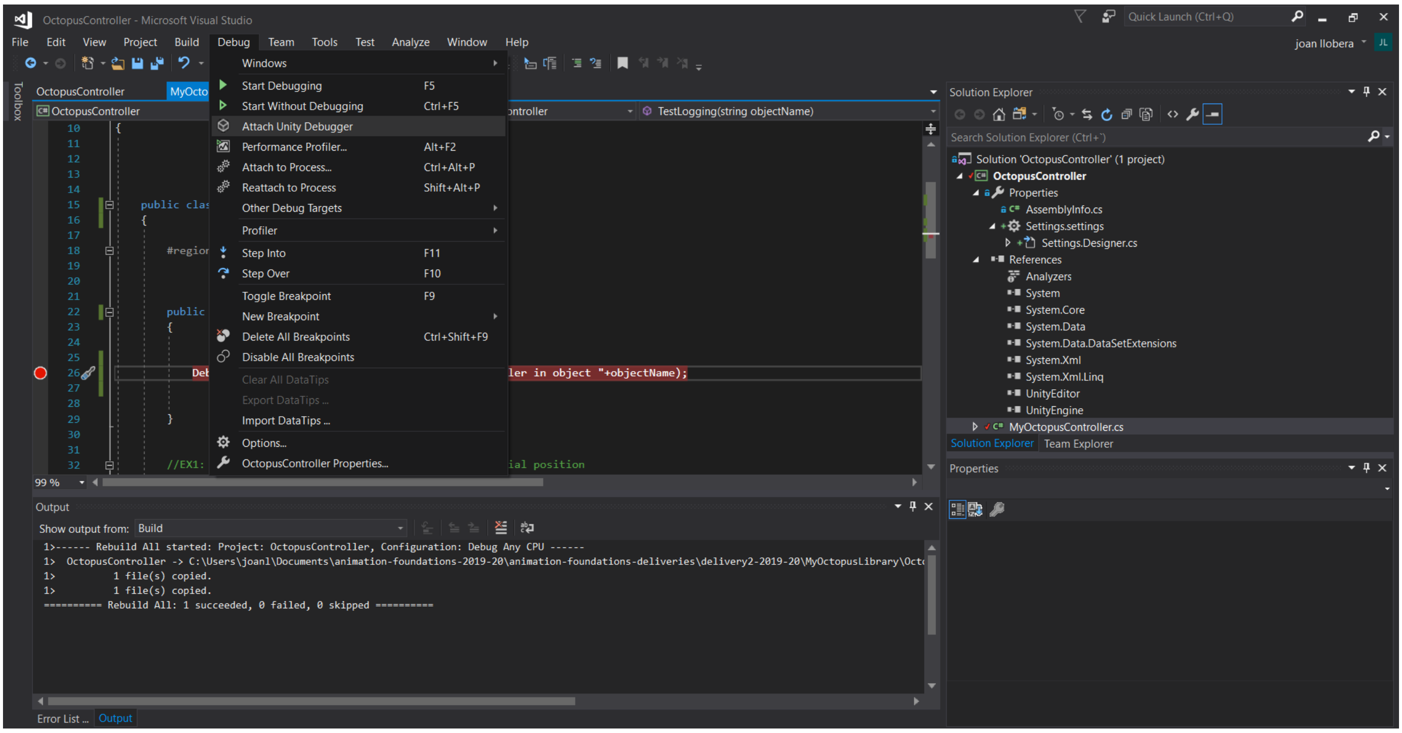
If you also want to debug the .dll using the Unity project within the Unity editor, the documentation is a bit confusing (see here: https://docs.unity3d.com/Manual/UsingDLL.html)

Something that has worked for me in the past is to also copy the .pdb inside the Plugin library. You can use the following post build instructions:

copy /Y "$(TargetDir)$(ProjectName).dll" "$(SolutionDir)\..\..\MyUnityProject\Assets\Plugins\$(ProjectName).dll"

copy /Y "$(TargetDir)$(ProjectName).pdb" "$(SolutionDir)\..\..\MyUnityProject\Assets\Plugins\$(ProjectName).pdb"

Then Attach the unity debugger (see screenshot below):



```
1>------ Rebuild All started: Project: OctopusController, Configuration: Debug Any CPU ------
1>  OctopusController -> C:\Users\joanl\Documents\animation-foundations-2019-20\animation-foundations-deliveries\delivery2-2019-20\MyOctopusLibrary\Octo
1>          1 file(s) copied.
1>          1 file(s) copied.
========== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==========
```

In the popup that follows select the Unity Editor version with which you want to debug, and finally press play in the editor if not done already.