# A better way to integrate Code and Unity

In the previous article we discussed how to integrate Unity and Visual Studio Code from scratch. But thanks to [Unity plugin](https://github.com/dotBunny/VSCode/) from @reapazor you can simplify this work.

As you know, it’s possible to use Unity plugins to extend Unity interface and execute almost everything. If you have source code of a Plugin you simply need to place it anywhere in the Assets folder and Unity will compile it on fly and use attributes there to extend the standard menu.

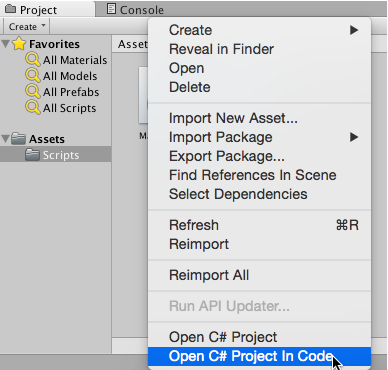
The VS Code plugin contains one source code file only, so, you can clone the full repository using

**git clone https://github.com/dotBunny/VSCode.git**

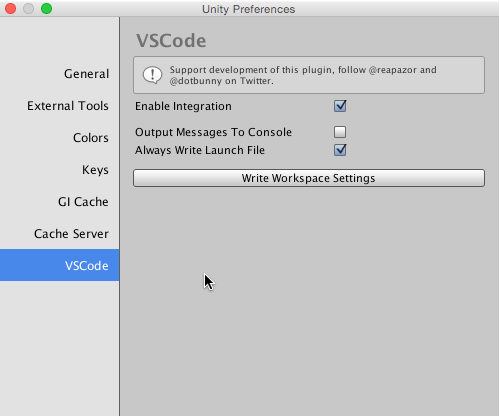
or extract the file itself from zip package.

In my experiments I am using Unity3D 5.2, so it’s not very important where to place this file. For example, you can place it as before using Plugins/Editor folder or you can select any other folder in your Assets tree.

Once you add the file to your Assets folder, Unity will compile it, and you can find new menu item using the context menu or Assets menu item.



Using Open C# Project in Code you can open Visual Studio Code and it will open the right catalog and select the right project by default. But the plugin supports much more things. In order to see al features there, let’s open Unity->Preferences and you will find VSCode there:



Let’s make a short overview to the options there:

* **Enable Integration** – if this option is checked, you can use VS Code as the default editor. This option should be on by default but you can change it any time instead by using **Unity->Preferences->External Tools** tab;
* **Always Write Launch File** – by default you cannot debug the project. In order to enable debugging you need to create launch.json file and configure it. But the plugin can create the launch.json for you. In order to do it you need to enable this feature, turning on the checkbox and run your project from Unity editor. The plugin will read information about the debugging port and update/create launch.json file;
* **Write Workspace Settings** – If you open your project in VS Code you can find lots of different files that you don’t want to see. It’s easy to fix, add your own settings for the workspace and exclude all not needed files. Using this button, you can do it automatically – just click it;

So, as you can see we spent less time than before in order to integrate Unity and VS Code using the plugin. And the plugin supports much more features by default than we showed in the previous article: correct context (file and line) when you are opening a file; preferences initialization; workspace settings; debugging.

I hope that you will enjoy using this plugin.