General Information

uIntelliSense brings all the power of IntelliSense code hints to your Unity code.

While coding, have you ever forgot the exact meaning of some method parameter, or stumbled upon a method you've never used before? The only option was to start your browser, open the Scripting Reference, find the type, find the method, and only then you'd finally be able to get what you've searched. But isn't that *way* too much work when all you *really* have to do is just hover the cursor over the point of interest? And that is what *uIntelliSense* is about, providing useful hints for the Unity API — descriptions for types, variables, methods, method parameters etc.

Once you'll try it, you'll get used to it immediately. It is extremely helpful and time-saving, especially if you are a beginner who doesn't know the Unity API very well.

Second part of *uIntelliSense* is a Visual Studio extension for *Visual Studio 2012* and newer, for even better experience: in-IDE integrated Scripting Reference browser, Quick Search for Unity API and more!

On macOS, *uIntelliSense* also downloads and installs missing documentation for non-Unity .NET classes to further improve the experience.

Unity versions 4.6.x – 5.x are supported, both Windows and macOS Unity editors. Pro license is not required.

Works with both MonoDevelop and Visual Studio. All Visual Studio versions are supported for IntelliSense hints, Visual Studio 2012 or newer is required for the extension.

Setup

Using *uIntelliSense* is as simple as possible. Open the *uIntelliSense* dialog by calling

Tools → Lost Polygon → uIntelliSense

Click *"Generate IntelliSense data"* and wait for the process to end (usually a minute or two).

You might be prompted with a UAC administrator privilege request dialog on Windows, or an administrator rights request on macOS. This is required because *uIntelliSense* has to create a few files inside you Unity installation. This can't harm neither your Unity installation nor your computer in any way — only a few files containing the documentation data are created.

Restart your IDE when the process is finished. That's it! You should be able to see the code hints now. Just hover the cursor over a Unity API type or type member.
Note that *uIntelliSense* uses Unity documentation, so it can't show anything that wasn't present in the documentation in the first place.

You can safely delete *uIntelliSense* directory from your project when the process is complete – hints are not tied to a specific project, they will be used automatically everywhere. The exception to this is if you are using *Visual Studio Tools for Unity* (formerly known as UnityVS) — you'll have to leave *uIntelliSense* directory in your project because of a workaround required for correctly managing projects created by *Visual Studio Tools for Unity*.

Click *"Install Visual Studio Extension"* button to install the Visual Studio Extension. You can read more about it in the next section.

Use the dropdown triangle menu to uninstall *uIntelliSense* data or visit the support forum thread.

Visual Studio Extension

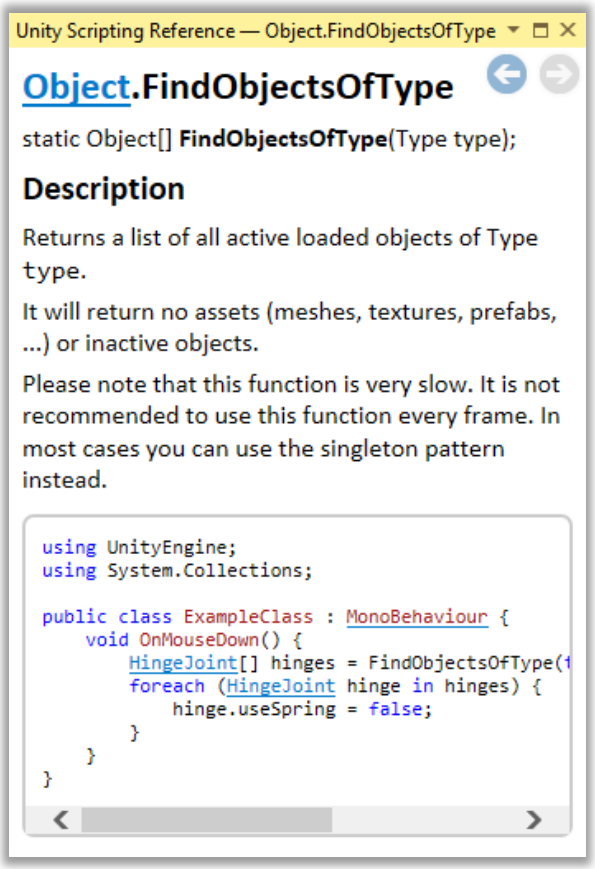
uIntelliSense includes a Visual Studio extension that improves your Unity workflow in Visual Studio and pretty much eliminates the need to open full Scripting Reference in your browser ever again!

To install the extension, click *"Install Visual Studio Extension"* button.

Extension is compatible with Visual Studio 2012 and newer.

Quick View Tool Window

Quick View tool window allows you to explore Scripting Reference without leaving Visual Studio. It is a regular Visual Studio window, so it can be dragged around and compactly docked anywhere you want.



You can open this window by using the following Visual Studio menu item:

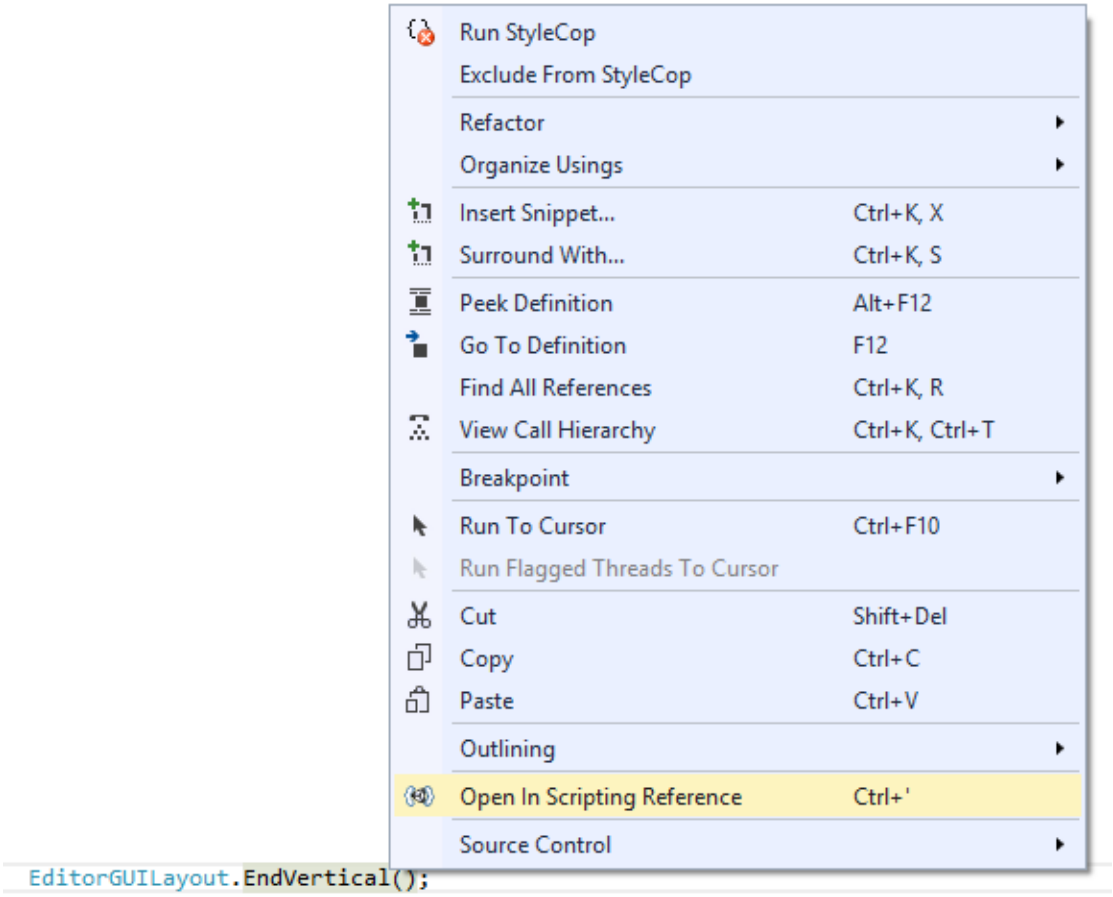
View → Other Windows → Unity Scripting Reference

Holding the *Shift* key while clicking a link will open it in your default browser.

Arrow keys in upper right corner of the window are used to navigate the browsing history. You can also use *Backspace, Shift+Backspace, Ctrl+Right/Left Arrow* shortcuts for the same purpose.

Using *Ctrl+Mouse* Wheel allows you to zoom in/out of the window content.

The common way of productively using the Quick View window is to place your cursor over some code symbol, open the context menu by right-clicking, and then click *"Open In Scripting Reference"*.



You can also use *Ctrl+'* (single quotation mark) shortcut to execute this command without using the context menu. If the default shortcut is not working for some reason, you can reassign it in

Tools → Options… → Environment → Keyboard

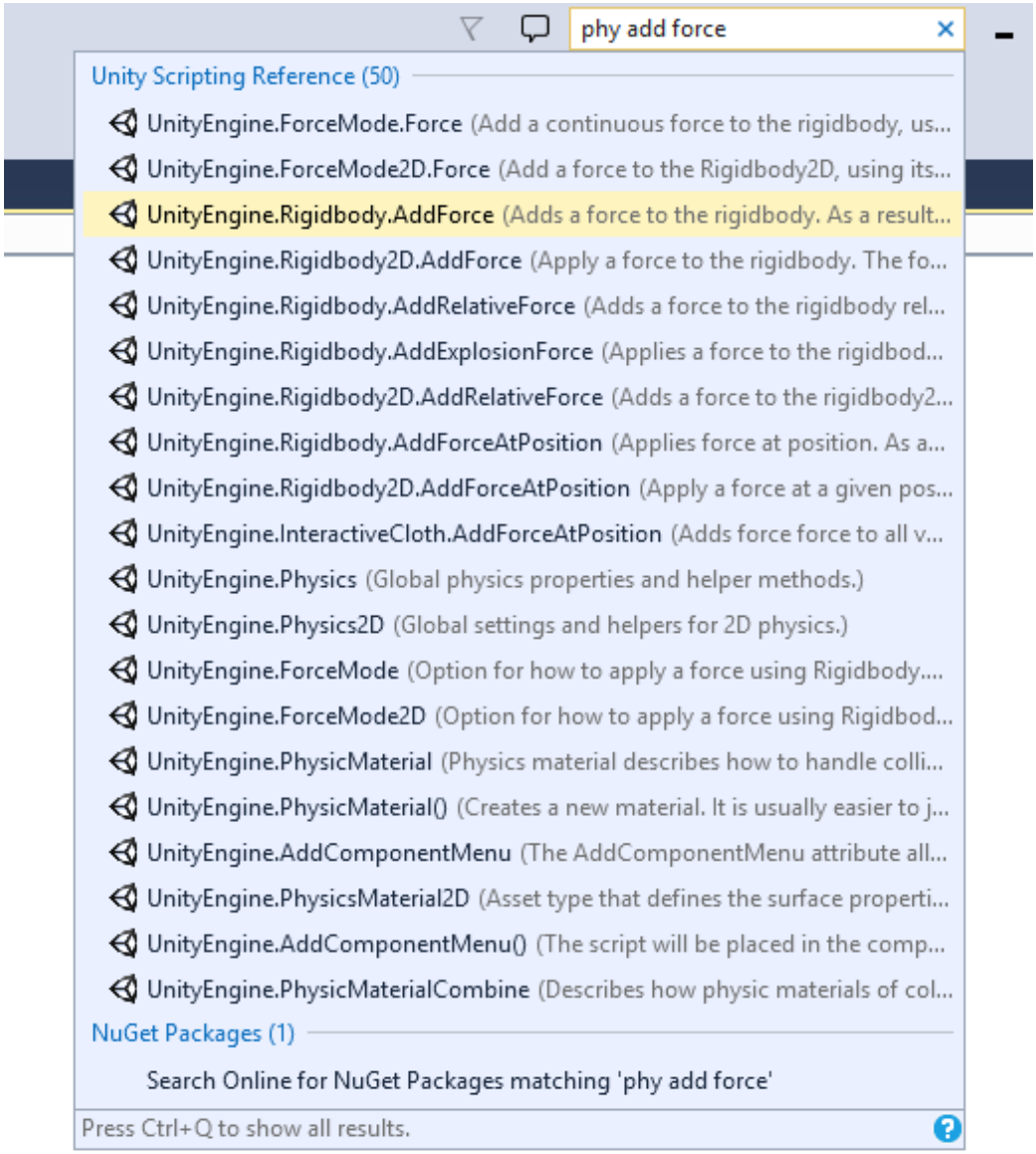
Use *"EditorContextMenus.CodeWindow.OpenInScriptingReference"* command name to find it quickly.

Pro-tip: the best way to use the Quick View tool window is to dock it to the left or right side of Visual Studio and enable Auto-hide on it. To do that, dock the window by dragging it to a side of the screen, and click the pin icon in the window title.

That way, the Quick View automatically shows up when you request it via Quick Search or *"Open In Scripting Reference"* context menu item, and hides whenever you click anywhere outside of it. The window won't also consume any screen space, and you can actually make it big enough for more comfortable reading.

Quick Launch Search

uIntelliSense Visual Studio Extension adds Scripting Reference search capabilities to the Quick Search field, located at the upper right corner of Visual Studio.



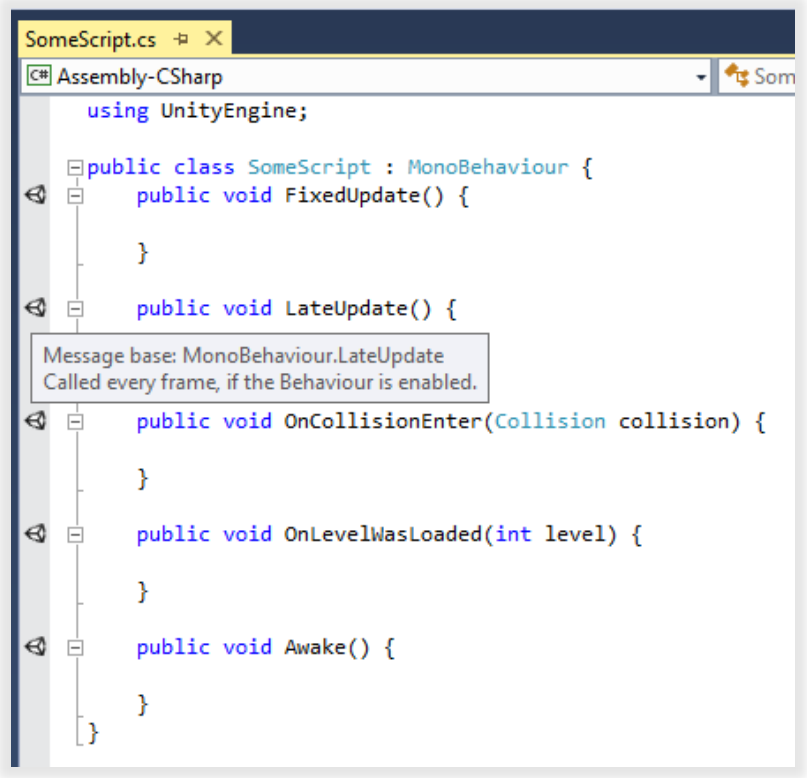
The search is smart enough to understand parts of words. For example, if you type "mat ort", the first result would be *UnityEngine.Matrix4x4.Ortho*.

Clicking on a search result will open it in the *Quick View* window. Clicking while holding *Shift* key will open the result in your default browser.

Use the *Ctrl+Q* shortcut to quickly focus on the Quick Search and start typing right away.

Unity Message Method Icons

uIntelliSense Visual Studio Extension can add icons to the code editor that show if the method is actually implementing a Unity message method like *Start*, *Update*, *OnCollisionEnter* etc. This is useful to visually separate these methods from regular ones, as they are not actually overrides of any base class methods, and that can be confusing sometimes.



Options

uIntelliSense Visual Studio Extension options can be found in

Tools → Options… → uIntelliSense

Default Unity root path:

If possible, uIntelliSense uses the Unity installation that was used to create the solution to get the best suitable Scripting Reference. However, if this fails for some reason, this path will be used instead. Generally, you don't need to modify this parameter unless you know what you are doing, as it is assigned automatically.

Use online documentation when opening links in browser:

When this option is enabled, clicking links and search results while holding *Shift* key will open the online Scripting Reference instead of local one.

Show icons for Unity message methods (Awake, Start, Update, etc.)

Enables/disables displaying Unity message method icons in the code editor.

Default zoom

Default zoom level to use for *Quick View* window contents.

Contact

For any questions about this plugin, feel free to contact me at:

Unity forums thread:	https://forum.unity3d.com/threads/uintellisense-code-hints-for-unity-api-released.276539/
E-mail:	contact@lostpolygon.com
Skype:	serhii.yolkin
Web-site:	http://lostpolygon.com