# 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 4.7.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).
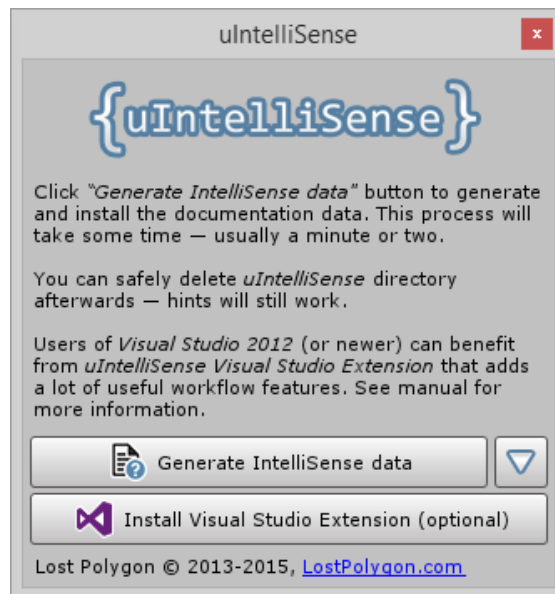


You might be prompted with a UAC administrator privilege request dialog on Windows, or an administrator rights request on macOS. This is required because *uIntelliSense* has to create a few files inside you Unity installation. This can't harm neither your Unity installation nor your computer in any way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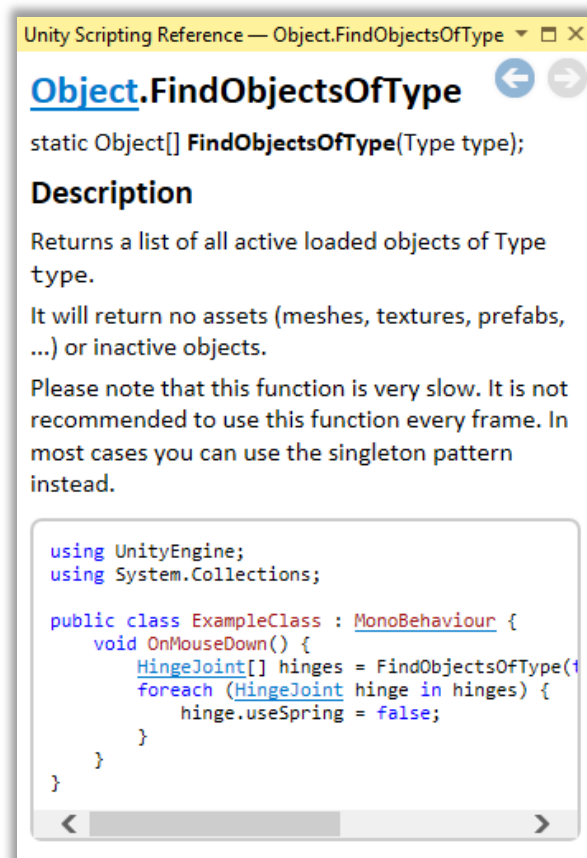


You can open this window by using the following Visual Studio menu item:

<p align="center"><em>View → Other Windows → Unity Scripting Reference</em></p>

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"*.
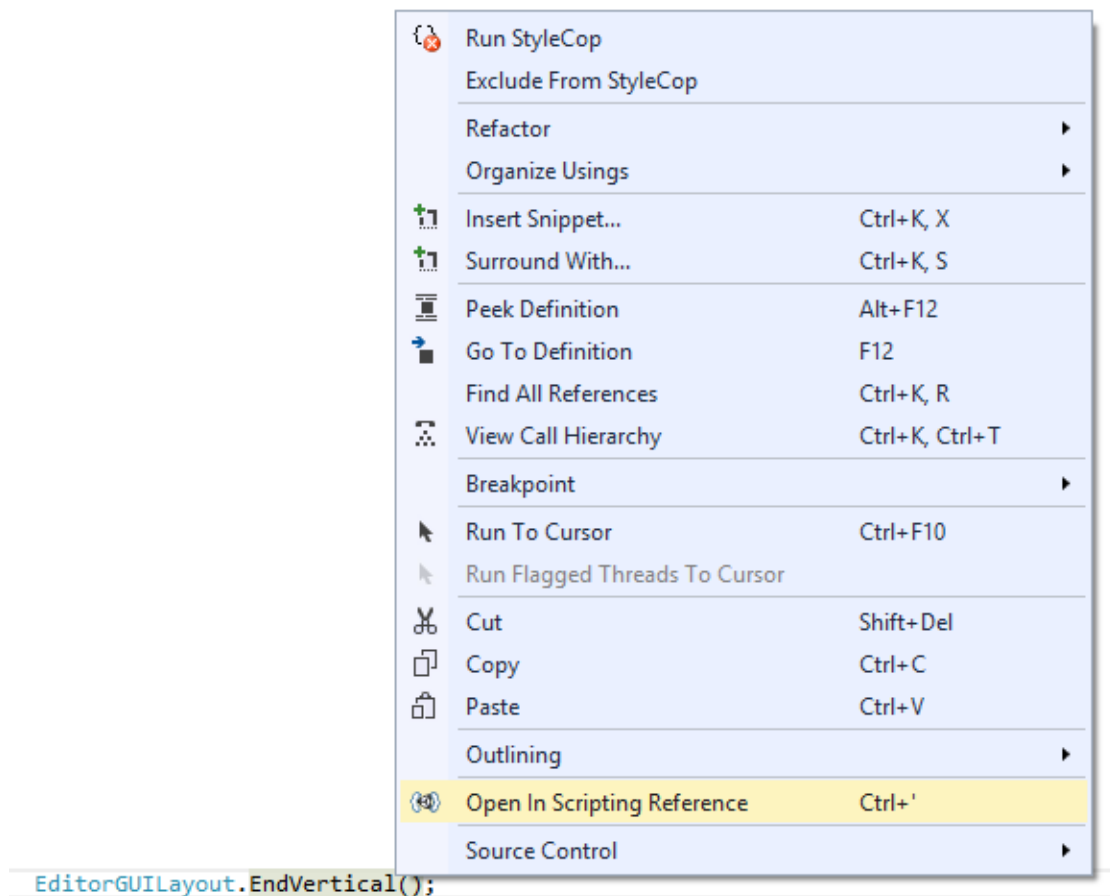


You can also use *Ctrl+'* (single quotation mark) shortcut to execute this command without using the context menu. If the default shortcut is not working for some reason, you can reassign it in

<div align="center">

*Tools → Options… → Environment → Keyboard*

</div>

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.
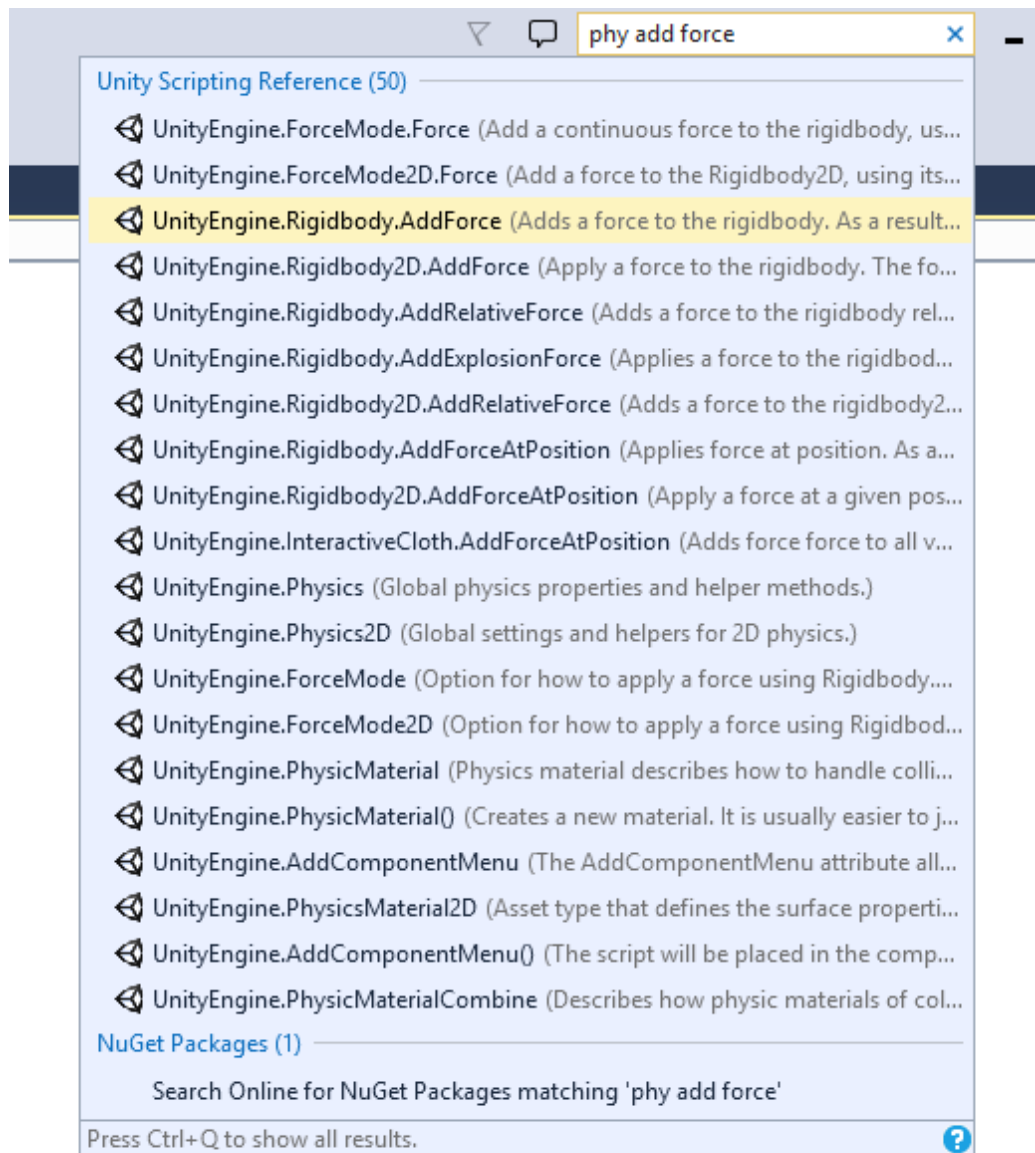


The search is smart enough to understand parts of words. For example, if you type "mat ort", the first result would be *UnityEngine.Matrix4x4.Ortho*.

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.
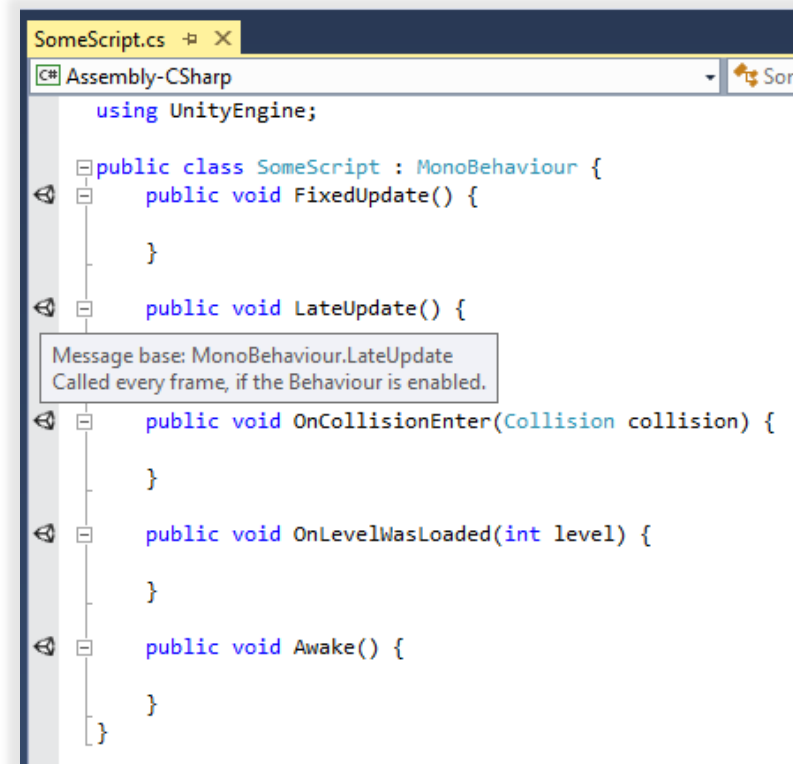


# Options

*uIntelliSense Visual Studio Extension* options can be found in

$$Tools \rightarrow Options\ldots \rightarrow uIntelliSense$$

**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 |



# Version history

**1.7.0:**
*Unity tool:*
- Unity 5.5 support.
- Fixed documentation data not being updated when Visual Studio Tools for Unity is used.
- Added support for EditorTestsRunner and HoloLens documentation.
- Added "Generate Narrow Tooltips" option that limits the size of a single tooltip line. Useful for Visual Studio.
- Minor fixes and improvements.

*Visual Studio extension:*
- Preliminary Visual Studio 2017 support.
- Improved Quick Search performance and accuracy.

**1.6.0:**
*Unity tool:*
- Added support for Unity IAP and Unity Ads documentation.
- Preliminary support for Unity Linux Editor.
- Menu items are now in a separate .cs file for customizability.
- Fixed original IntelliSense XML files not backing up properly on Unity 5.2+.
- Minor fixes and improvements.

*Visual Studio extension:*
- Added Unity message methods icons for *AssetPostprocessor, AssetModificationProcessor, ScriptableWizard*.
- Fixed error spam when not able to retrieve the base class of the class.
- Fixed Unity message methods with parameters of Unity types being unavailable.
- Fixed *"Open In Scripting Reference"* not working in Visual Studio 2012 and 2013 for Unity message methods.
- Improved Quick Search ranking algorithm.
- Minor fixes and improvements.

**1.5.1:**
*Visual Studio extension:*
- Fixed Unity message methods with parameters of Unity types being unavailable.
- Fixed Ctrl+' not working again after being used on non-Unity symbol.

**1.5.0:**
*Unity tool:*
- Improved compatibility with Unity 5.2.
- Parsing improvements, generates cleaner data now.
- Greatly improved handling of UAC on Windows and users with limited access on macOS.
- Fixed "Uninstall uIntelliSense data" button always being enabled on Unity 5.1 and newer.
- Original documentation data is now automatically backed up when generating uIntelliSense documentation and restored when deleting (Unity 5.1 and newer).
- Fixed error message spam caused by Visual Studio locking the documentation data files when *Visual Studio Tools for Unity* is used.
- Fixed being unable to install documentation data if current project had path with Unicode symbols in it.
- Generally better handling of errors and some edge cases.

*Visual Studio extension:*
- Fixed invisible labels in extension settings.
- Fixed crash when opening a document that has no path.
- Automatically scale the Quick View window according to system DPI settings.
- Added support for Back/Forward mouse buttons and *Ctrl+Left/Right Arrow* in the Quick View tool window for navigation.
- Quick View window can't now be used as a drag-n-drop target.
- Fixed some color scheme issues in Quick View tool window.
- Fixed infinite loop when base class of the class that is currently edited couldn't be detected.
- Added *MonoBehaviour.OnTransformParentChanged* and *MonoBehaviour.OnTransformChildrenChanged* to the list of message methods.
- *"Open In Scripting Reference"* now works with Unity message methods.

- *"Open In Scripting Reference"* now opens type documentation if no documentation for constructor was found.
- Added "Default zoom" value to the settings, allows setting the initial zoom in Quick View tool window.
- Increased search results count limit from 50 to 75 in Quick Search.
- Performance improvements.

**1.4.3:**
- Fixed Unity 5.1 compatibility issues.
- Added support for UNet and Analytics documentation.
- Stability improvements in Visual Studio Extension.

**1.4.2:**
- Fixed Unity 5.0.1 compatibility issue.
- Fixed a regression in Visual Studio Extension that made default Internet Explorer context menu show up instead of a custom one in Quick View window.
- Stability improvements in Visual Studio Extension.

**1.4.1:**
- Fixed an issue that lead to some files not being copied when UAC is active.

**1.4.0:**
- Improved Unity 5 support.
- Added downloading of .NET classes documentation for macOS.
- Added theme support for Visual Studio Extension. It automatically adapts to the current Visual Studio theme.
- Visual Studio Extension now saves the scrolling position when navigating in history or refreshing the page.
- Improved operation of Visual Studio Extension when working on projects generated by Visual Studio Tools for Unity.
- Removed navigation sounds in Visual Studio Extension when Internet Explorer 8 is installed in the system.
- Adjusted security policy, Internet Explorer information bar should not appear anymore.
- Stability improvements.

**1.3.0.3:**
- Stability improvements.

**1.3.0.2:**
- Added support for using *Visual Studio Tools for Unity* with Visual Studio in a virtual machine on macOS.

**1.3.0.1:**
- Fixed a rare crash in Visual Studio Extension when parsing page without Boo examples.

**1.3.0:**
- Improved documentation parsing and generation.
- Improved UI workflow of Visual Studio Extension.
- Fixed some rare crashes in Visual Studio Extension.
- Fixed Visual Studio Extension crashing when Internet Explorer 8 was used in the system.
- Fixed a possible crash on Unity 4.5.

**1.2.0:**
- Major parsing improvements. Much more data is now retrieved.
- Added Visual Studio extension with various documentation-related features.
- Added `<exception>` and `<seealso>` elements support in XML documentation data.
- Improved `<see>` element handling.
- Added an option to uninstall uIntelliSense data.
- Improved compatibility with User Access Control on Windows.
- UI improvements.
- Major internal code improvements for better operation.

**1.1.2:**
- Fixed a rare NullReferenceException on macOS.

**1.1.1:**
- Minor interface fixes and changes.

**1.1.0:**
- Added experimental Unity 5 Beta support (tested on Beta 14).

**1.0.2:**
- Fixed a rare issue when menu wasn't showing up after importing uIntelliSense into an existing project.

**1.0.1:**
- Fixed generator window being shown garbled on some machines.
- UnityVS compatibility wasn't actually working. This is fixed now.
- Fixed some minor parsing bugs.

**1.0:**
- Initial release.