

# DOWNLOAD

Visual Studio For Mac 2015

Download Microsoft Visual Studio 2015 for free. Microsoft Visual Studio 2015 - Visual Studio 2015 is an integrated suite of developer productivity tools, cloud services, and extensions that enable you and your team to create great apps and games for the web, for Windows Store, for the desktop, for Android, and for iOS. In Visual Studio for Mac visual studio for mac macos 10.15 Fixed - Pending Release Jonathan Peppers MSFT commented Sep 15, '20 38 Votes. The NuGet Package Manager UI in Visual Studio on Windows allows you to easily install, uninstall, and update NuGet packages in projects and solutions. For the experience in Visual Studio for Mac, see Including a NuGet package in your project. The Package Manager UI is not included with Visual Studio Code.

- [Visual Studio For Mac 2019 Download](#)
- [Visual Studio 2015 For Mac](#)
- [Visual Studio Community For Mac](#)

-->

Python is a popular programming language that is reliable, flexible, easy to learn, free to use on all operating systems, and supported by both a strong developer community and many free libraries. Python supports all manners of development, including web applications, web services, desktop apps, scripting, and scientific computing, and is used by many universities, scientists, casual developers, and professional developers alike. You can learn more about the language on [python.org](http://python.org) and Python for Beginners.

Visual Studio is a powerful Python IDE on Windows. Visual Studio provides open-source support for the Python language through the **Python Development** and **Data Science** workloads (Visual Studio 2017 and later) and the free Python Tools for Visual Studio extension (Visual Studio 2015 and earlier).

Python is not presently supported in Visual Studio for Mac, but is available on Mac and Linux through Visual Studio Code (see questions and answers).

To get started:

- Follow the installation instructions to set up the Python workload.
- Familiarize yourself with the Python capabilities of Visual Studio through the sections in this article.
- Go through one or more of the Quickstarts to create a project. If you're unsure, start with Create a web app with Flask.
- Go through one or more of the Quickstarts to create a project. If you're unsure, start with Quickstart: Open and run Python code in a folder or Create a web app with Flask.
- Follow the Work with Python in Visual Studio tutorial for a full end-to-end experience.

Note

Visual Studio supports Python version 2.7, as well as version 3.5 through 3.7. While it is possible to use Visual Studio to edit code written in other versions of Python, those versions are not officially supported and features such as IntelliSense and debugging might not work. Python version 3.8 support is still under development, specific details about support can be seen in this tracking issue on GitHub.

## Support for multiple interpreters

Visual Studio's **Python Environments** window (shown below in a wide, expanded view) gives you a single place to manage all of your global Python environments, conda environments, and virtual environments. Visual Studio automatically detects installations of Python in standard locations, and allows you to configure custom installations. With each environment, you can easily manage packages, open an interactive window for that environment, and access environment folders.

Use the **Open interactive window** command to run Python interactively within the context of Visual Studio. Use the **Open in PowerShell** command to open a separate command window in the folder of the selected environment. From that command window you can run any python script.

For more information:

## Rich editing, IntelliSense, and code comprehension

Visual Studio provides a first-class Python editor, including syntax coloring, auto-complete across all your code and libraries, code formatting, signature help, refactoring, linting, and type hints. Visual Studio also provides unique features like class view, **Go to Definition**, **Find All References**, and code snippets. Direct integration with the Interactive window helps you quickly develop Python code that's already saved in a file.

For more information:

- Docs: Edit Python code
- Docs: Format code
- Docs: Refactor code
- Docs: Use a linter
- General Visual Studio feature docs: Features of the code editor

## Interactive window

For every Python environment known to Visual Studio, you can easily open the same interactive (REPL) environment for a Python interpreter directly within Visual Studio, rather than using a separate command prompt. You can easily switch between environments as well. (To open a separate command prompt, select your desired environment in the **Python Environments** window, then select the **Open in PowerShell** command as explained earlier under Support for multiple interpreters.)

Visual Studio also provides tight integration between the Python code editor and the **Interactive** window. The **Ctrl+Enter** keyboard shortcut conveniently sends the current line of code (or code block) in the editor to the **Interactive** window, then moves to the next line (or block). **Ctrl+Enter** lets you easily step through code without having to run the debugger. You can also send selected code to the **Interactive** window with the same keystroke, and easily paste code from the **Interactive** window into the editor.

Together, these capabilities allow you to work out details for a segment of code in the **Interactive** window and easily save the results in a file in the editor.

Visual Studio also supports IPython/Jupyter in the REPL, including inline plots, .NET, and Windows Presentation Foundation (WPF).

For more information:

## Project system, and project and item templates

Note

Visual Studio 2019 supports opening a folder containing Python code and running that code without creating Visual Studio project and solution files. For more information, see Quickstart: Open and run Python code in a folder. There are, however, benefits to using a project file, as explained in this section.

Visual Studio helps you manage the complexity of a project as it grows over time. A *Visual Studio project* is much more than a folder structure: it includes an understanding of how different files are used and how they relate to each other. Visual Studio helps you distinguish app code, test code, web pages, JavaScript, build scripts, and so on, which then enable file-appropriate features. A Visual Studio solution, moreover, helps you manage multiple related projects, such as a Python project and a C++ extension project.

Project and item templates automate the process of setting up different types of projects and files, saving you valuable time and relieving you from managing intricate and error-prone details. Visual Studio provides templates for web, Azure, data science, console, and other types of projects, along with templates for files like Python classes, unit tests, Azure web configuration, HTML, and even Django apps.

For more information:

- Docs: Manage Python projects
- Docs: Item templates reference
- Docs: Python project templates
- Docs: Work with C++ and Python
- General Visual Studio feature docs: Project and item templates
- General Visual Studio feature docs: Solutions and projects in Visual Studio

## Full-featured debugging

One of Visual Studio's strengths is its powerful debugger. For Python in particular, Visual Studio includes Python/C++ mixed-mode debugging, remote debugging on Linux, debugging within the **Interactive** window, and debugging Python unit tests.

In Visual Studio 2019, you can run and debug code without having a Visual Studio project file. See Quickstart: Open and run Python code in a folder for an example.

For more information:

- Docs: Debug Python
- Docs: Python/C++ mixed-mode debugging
- Docs: Remote debugging on Linux
- General Visual Studio feature docs: Feature tour of the Visual Studio Debugger



IronPython WPF (with XAML designer)	✓	✓	✓	✓	✓	✓	✓	✓	✓
IronPython Windows Forms	✓	✓	✓	✓	✓	✓	✓	✓	✓

Web	2017+	2015	2013 Comm	2013 Desktop	2013 Web	2013 Pro+	2012 Pro+	2010 SP1 Pro+
Django web project	✓	✓	✓	X	✓	✓	✓	✓
Bottle web project	✓	✓	✓	X	✓	✓	✓	✓
Flask web project	✓	✓	✓	X	✓	✓	✓	✓
Generic web project	✓	✓	✓	X	✓	✓	✓	✓

Azure	2017+	2015	2013 Comm	2013 Desktop	2013 Web	2013 Pro+	2012 Pro+	2010 SP1 Pro+
Deploy to web site	✓	✓	✓	X	✓	✓	✓	✓ <sup>2</sup>
Deploy to web role	✓	✓	✓	X	✓ <sup>4</sup>	✓ <sup>4</sup>	✓ <sup>3</sup>	X
Deploy to worker role	?	?	?	X	✓ <sup>4</sup>	✓ <sup>4</sup>	✓ <sup>3</sup>	X
Run in Azure emulator	?	?	?	X	✓ <sup>4</sup>	✓ <sup>4</sup>	✓ <sup>3</sup>	X
Remote debugging	✓	✓	✓	X	✓ <sup>5</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	X
Attach Server Explorer	✓	✓	✓	X	✓ <sup>7</sup>	✓ <sup>7</sup>	X	X

Django templates	2017+	2015	2013 Comm	2013 Desktop	2013 Web	2013 Pro+	2012 Pro+	2010 SP1 Pro+
Debugging		✓	✓	✓	X	✓	✓	✓
Auto-complete		✓	✓	✓	X	✓ <sup>5</sup>	✓ <sup>5</sup>	✓
Auto-complete for CSS and JavaScript		✓	✓	✓	X	✓ <sup>5</sup>	✓ <sup>5</sup>	X

Debugging	2017+	2015	2013 Comm	2013 Desktop	2013 Web	2013 Pro+	2012 Pro+	2010 SP1 Pro+
Debugging		✓	✓	✓	✓	✓	✓	✓
Debugging without a project		✓	✓	✓	✓	✓	✓	✓
Debugging - attach to editing		✓	✓	✓	✓	X	✓	✓
Mixed-mode debugging		✓	✓	✓	✓	✓	✓	X
Remote debugging (Windows, Mac OS X, Linux)		✓	✓	✓	✓	X	✓	✓
Debug Interactive window		✓	✓	✓	✓	✓	✓	✓

Profiling	2017+	2015	2013 Comm	2013 Desktop	2013 Web	2013 Pro+	2012 Pro+	2010 SP1 Pro+
Profiling	✓	✓	✓	X	X	✓	✓	✓

Test	2017+	2015	2013 Comm	2013 Desktop	2013 Web	2013 Pro+	2012 Pro+	2010 SP1 Pro+
Test explorer	✓	✓	✓	✓	✓	✓	✓	X
Run test	✓	✓	✓	✓	✓	✓	✓	X
Debug test	✓	✓	✓	✓	✓	✓	✓	X

- 
- Git support for Visual Studio 2012 is available in the Visual Studio Tools for Git extension, available on the Visual Studio Marketplace.
- 
- Deployment to Azure Web Site requires Azure SDK for .NET 2.1 - Visual Studio 2010 SP1. Later versions don't support Visual Studio 2010.
- 
- Support for Azure Web Role and Worker Role requires Azure SDK for .NET 2.3 - VS 2012 or later.
- 
- Support for Azure Web Role and Worker Role requires Azure SDK for .NET 2.3 - VS 2013 or later.
- 
- Django template editor in Visual Studio 2013 has some known issues that are resolved by installing Update 2.
- 
- Requires Windows 8 or later. Visual Studio 2013 Express for Web doesn't have the **Attach to Process** dialog, but Azure Web Site remote debugging is still possible using the **Attach Debugger (Python)** command in **Server Explorer**. Remote debugging requires Azure SDK for .NET 2.3 - Visual Studio 2013 or later.
- 
- Requires Windows 8 or later. **Attach Debugger (Python)** command in **Server Explorer** requires Azure SDK for .NET 2.3 - Visual Studio 2013 or later.
- 
- Requires Windows 8 or later.

-->

You can use this guide to uninstall each component in Visual Studio for Mac individually by navigating to the relevant section, or you can use the scripts provided in the Uninstall Script section to uninstall everything.

Note

This information will only remove Visual Studio 2019 or 2017 for Mac from your machine. to uninstall Visual Studio Code, see this issue for details.

## Uninstall Script

There are two scripts that can be used to uninstall Visual Studio for Mac and all components for your machine:

The following sections provide information on downloading and using the scripts.

### Visual Studio for Mac and Xamarin script

You can uninstall Visual Studio and Xamarin components in one go by using the uninstall script.

This uninstall script contains most of the commands that you will find in the article. There are three main omissions from the script and are not included due to possible external dependencies. To remove this, jump to the relevant section below and remove them manually:

To run the script, do the following steps:

### Visual Studio For Mac 2019 Download

- 
- Right-click on the script and select **Save As** to save the file on your Mac.
- 
- Open Terminal and change the working directory to where the script was downloaded:
- 
- Make the script executable and the run it with **sudo**:
-

8. Finally, delete the uninstall script and remove Visual Studio for Mac from the dock (if it's there).

### [.NET Core script](#)

The uninstall script for .NET Core is located in the dotnet cli repo

To run the script, do the following steps:

- 1.
2. Right-click on the script and select **Save As** to save the file on your Mac.
- 3.
4. Open Terminal and change the working directory to where the script was downloaded:
- 5.
6. Make the script executable and the run it with **sudo**:
- 7.
8. Finally, delete the .NET Core uninstall script.

### [Uninstall Visual Studio for Mac](#)

The first step in uninstalling Visual Studio from a Mac is to locate **Visual Studio.app** in the **/Applications** directory and drag it to the **Trash Can**. Alternatively, right-click and select **Move to Trash** as illustrated in the following image:

Deleting this app bundle removes Visual Studio for Mac, even though there may be other files related to Xamarin still on the file system.

To remove all traces of Visual Studio for Mac, run the following commands in Terminal:

You may also want to remove the following directory containing various Xamarin files and folders. However, before you do you should be aware that this directory contains the Android signing keys. For more information refer to the section **Uninstalling Android SDK and Java SDK**:

### [Uninstall Mono SDK \(MDK\)](#)

Mono is an open-source implementation of Microsoft's .NET Framework and is used by all Xamarin Products—Xamarin.iOS, Xamarin.Android, and Xamarin.Mac to allow development of these platforms in C#.

Warning

There are other applications outside of Visual Studio for Mac that also use Mono, such as Unity.Be sure that there are no other dependencies on Mono before uninstalling it.

To remove the Mono Framework from a machine, run the following commands in Terminal:

### [Uninstall Xamarin.Android](#)

There are a number of items required for the installation and use of Xamarin.Android, such as the Android SDK and Java SDK.

Use the following commands to remove Xamarin.Android:

### [Uninstall Android SDK and Java SDK](#)

The Android SDK is required for development of Android applications. To completely remove all parts of the Android SDK, locate the file at **~/Library/Developer/Xamarin/** and move it to **Trash**.

Warning

### [Visual Studio 2015 For Mac](#)

You should be aware that Android signing keys that are generated by Visual Studio for Mac are located in **~/Library/Developer/Xamarin/Keystore**. Make sure to back these up appropriately, or avoid removing this directory if you wish to keep your keystore.

The Java SDK (JDK) does not need to be uninstalled, as it is already pre-packaged as part of Mac OS X / macOS.

### [Uninstall Android AVD](#)

Warning

There are other applications outside of Visual Studio for Mac that also use Android AVD and these additional android components, such as Android Studio.Removing this directory may cause projects to break in Android Studio.

To remove any Android AVDs and additional Android components use the following command:

To remove only the Android AVDs use the following command:

### [Uninstall Xamarin.iOS](#)

Xamarin.iOS allows iOS application development using C# or F# with Visual Studio for Mac.

Use the following commands in Terminal to remove all Xamarin.iOS files from a file system:

### [Uninstall Xamarin.Mac](#)

Xamarin.Mac can be removed from your machine using the following two commands to eradicate the product and license from your Mac respectively:

### [Visual Studio Community For Mac](#)

#### [Uninstall Workbooks and Inspector](#)

Starting with 1.2.2, Xamarin Workbooks & Inspector can be uninstalled from a terminal by running:

For older versions, you need to manually remove the following artifacts:

- Delete the Workbooks app at **'/Applications/Xamarin Workbooks.app'**
- Delete the Inspector app at **'Applications/Xamarin Inspector.app'**
- Delete the add-ins: **'~/Library/Application Support/XamarinStudio-6.0/LocalInstall/Addins/Xamarin.Interactive'** and **'~/Library/Application Support/XamarinStudio-6.0/LocalInstall/Addins/Xamarin.Inspector'**
- Delete Inspector and supporting files here: **/Library/Frameworks/Xamarin.Interactive.framework** and **/Library/Frameworks/Xamarin.Inspector.framework**

#### [Uninstall the Xamarin Profiler](#)

#### [Uninstall the Visual Studio Installer](#)

Use the following commands to remove all traces of the Xamarin Universal Installer:

### [Uninstall Visual Studio 2019 for Mac Preview](#)

Visual Studio 2019 for Mac Preview was launched as a separate preview, allowing you to continue to work with your Visual Studio 2017 for Mac install side-by-side.

Now that Visual Studio 2019 for Mac has been released, you can now safely remove the Visual Studio 2019 for Mac Preview application.

To uninstall the preview application bundle, select **Visual Studio (Preview)** from your **Applications** folder and click **Move to Trash**, as depicted in the following image:

You can also remove the Preview plist file with the following command:

### [See also](#)