Visual Studio 2012 For Mac Free Download

-->

- [Visual Studio 2015 For Mac Free Download](#)
- [Visual Studio 2012 Free Edition](#)

Microsoft visual studio 2012 free download - Visual Studio Professional 2012, Microsoft Visual Studio 2010 Ultimate, Microsoft Visual Studio 2010 Professional, and many more programs. For instructions on installing and updating Visual Studio 2019 for Mac, see the Install Visual Studio for Mac guide. To learn more about other related downloads, see the Downloads page. What's New in Visual Studio 2019 for Mac Visual Studio 2019 for Mac Releases. October 27, 2020 - Visual Studio 2019 for Mac version 8.7.9.

In this article, you'll learn how to install .NET Core on Windows. .NET Core is made up of the runtime and the SDK. The runtime is used to run a .NET Core app and may or may not be included with the app. The SDK is used to create .NET Core apps and libraries. The .NET Core runtime is always installed with the SDK.

The latest version of .NET Core is 3.1.

Supported releases

The following table is a list of currently supported .NET Core releases and the versions of Windows they're supported on. These versions remain supported until either the version of .NET Core reaches end-of-support or the version of Windows reaches end-of-life.

Windows 10 versions end-of-service dates are segmented by edition. Only **Home**, **Pro**, **Pro Education**, and **Pro for Workstations** editions are considered in the following table. Check the Windows lifecycle fact sheet for specific details.

- A ✓⬜ indicates that the version of Windows or .NET Core is still supported.
- A ✖ indicates that the version of Windows or .NET Core isn't supported on that Windows release.
- When both a version of Windows and a version of .NET Core have ✓⬜, that OS and .NET combination are supported.

Operating System	.NET Core 2.1	.NET Core 3.1	.NET 5 Preview
⬜⬜ Windows 10, Version 2004	✓⬜ 2.1	✓⬜ 3.1	✓⬜ 5.0 Preview
✓⬜ Windows 10, Version 1909	✓⬜ 2.1	✓⬜ 3.1	✓⬜ 5.0 Preview
✓⬜ Windows 10, Version 1903	✓⬜ 2.1	✓⬜ 3.1	✓⬜ 5.0 Preview
✓⬜ Windows 10, Version 1809	✓⬜ 2.1	✓⬜ 3.1	✓⬜ 5.0 Preview
✖ Windows 10, Version 1803	✓⬜ 2.1	✖ 3.1	✖ 5.0 Preview
✖ Windows 10, Version 1709	✖ 2.1	✖ 3.1	✖ 5.0 Preview
✖ Windows 10, Version 1703	✖ 2.1	✖ 3.1	✖ 5.0 Preview
✖ Windows 10, Version 1607	✖ 2.1	✖ 3.1	✖ 5.0 Preview
✖ Windows 10, Version 1511	✖ 2.1	✖ 3.1	✖ 5.0 Preview
✖ Windows 10, Version 1507	✖ 2.1	✖ 3.1	✖ 5.0 Preview

Unsupported releases

The following versions of .NET Core are ✖ no longer supported. The downloads for these still remain published:

- 3.0
- 2.2
- 2.0

Runtime information

The runtime is used to run apps created with .NET Core. When an app author publishes an app, they can include the runtime with their app. If they don't include the runtime, it's up to the user to install the runtime.

There are three different runtimes you can install on Windows:

ASP.NET Core runtime
Runs ASP.NET Core apps. Includes the .NET Core runtime.

Desktop runtime
Runs .NET Core WPF and .NET Core Windows Forms desktop apps for Windows. Includes the .NET Core runtime.

.NET Core runtime
This runtime is the simplest runtime and doesn't include any other runtime. It's highly recommended that you install both *ASP.NET Core runtime* and *Desktop runtime* for the best compatibility with .NET Core apps.
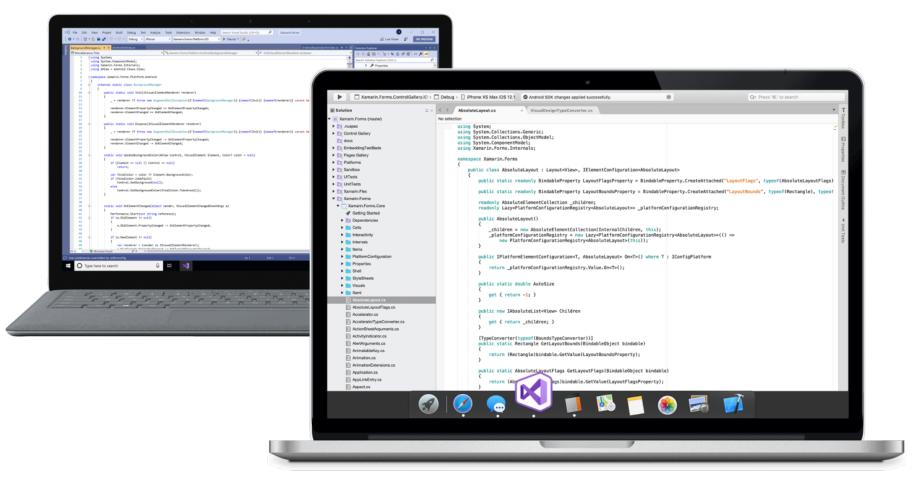
SDK information

The SDK is used to build and publish .NET Core apps and libraries. Installing the SDK includes all three runtimes: ASP.NET Core, Desktop, and .NET Core.

Dependencies

The following Windows versions are supported with .NET Core 3.1:

OS	Version	Architectures
Windows Client	7 SP1+, 8.1	x64, x86
Windows 10 Client	Version 1609+	x64, x86
Windows Server	2012 R2+	x64, x86
Nano Server	Version 1803+	x64, ARM32

For more information about .NET Core 3.1 supported operating systems, distributions, and lifecycle policy, see .NET Core 3.1 Supported OS Versions.

.NET Core 3.0 is currently out of support. For more information, see the .NET Core Support Policy.

The following Windows versions are supported with .NET Core 3.0:

Note

A + symbol represents the minimum version.

OS	Version	Architectures
Windows Client	7 SP1+, 8.1	x64, x86
Windows 10 Client	Version 1607+	x64, x86
Windows Server	2012 R2+	x64, x86
Nano Server	Version 1803+	x64, ARM32

For more information about .NET Core 3.0 supported operating systems, distributions, and lifecycle policy, see .NET Core 3.0 Supported OS Versions.

.NET Core 2.2 is currently out of support. For more information, see the .NET Core Support Policy.

The following Windows versions are supported with .NET Core 2.2:

OS	Version	Architectures
Windows Client	7 SP1+, 8.1	x64, x86
Windows 10 Client	Version 1607+	x64, x86
Windows Server	2008 R2 SP1+	x64, x86
Nano Server	Version 1803+	x64, ARM32

For more information about .NET Core 2.2 supported operating systems, distributions, and lifecycle policy, see .NET Core 2.2 Supported OS Versions.

The following Windows versions are supported with .NET Core 2.1:

Note

A + symbol represents the minimum version.

OS	Version	Architectures
Windows Client	7 SP1+, 8.1	x64, x86
Windows 10 Client	Version 1607+	x64, x86
Windows Server	2008 R2 SP1+	x64, x86
Nano Server	Version 1803+	x64,

For more information about .NET Core 2.1 supported operating systems, distributions, and lifecycle policy, see .NET Core 2.1 Supported OS Versions.

Windows 7 / Vista / 8.1 / Server 2008 R2 / Server 2012 R2

Additional dependencies are required if you're installing the .NET SDK or runtime on the following Windows versions:

- ✗ Windows 7 SP1
- ✗ Windows Vista SP 2
- ✓⬜ Windows 8.1
- ✓⬜ Windows Server 2008 R2
- ✓⬜ Windows Server 2012 R2

Install the following:

- Microsoft Visual C++ 2015 Redistributable Update 3.

The requirements above are also required if you come across one of the following errors:

The program can't start because *api-ms-win-crt-runtime-l1-1-0.dll* is missing from your computer. Try reinstalling the program to fix this problem.

- or -

The program can't start because *api-ms-win-cor-timezone-l1-1-0.dll* is missing from your computer. Try reinstalling the program to fix this problem.

- or -

The library *hostfxr.dll* was found, but loading it from *C:<path_to_app>hostfxr.dll* failed.

Install with PowerShell automation

The dotnet-install scripts are used for CI automation and non-admin installs of the runtime. You can download the script from the dotnet-install script reference page.

The script defaults to installing the latest long term support (LTS) version, which is .NET Core 3.1. You can choose a specific release by specifying the Channel switch. Include the Runtime switch to install a runtime. Otherwise, the script installs the SDK.

Install the SDK by omitting the -Runtime switch. The -Channel switch is set in this example to Current, which installs the latest supported version.

Install with Visual Studio

If you're using Visual Studio to develop .NET Core apps, the following table describes the minimum required version of Visual Studio based on the target .NET Core SDK version.

.NET Core SDK version	Visual Studio version
3.1	Visual Studio 2019 version 16.4 or higher.
3.0	Visual Studio 2019 version 16.3 or higher.

| 2.2 | Visual Studio 2017 version 15.9 or higher. |
| 2.1 | Visual Studio 2017 version 15.7 or higher. |

If you already have Visual Studio installed, you can check your version with the following steps.

1. Open Visual Studio.
2. Select **Help** > **About Microsoft Visual Studio**.
3. Read the version number from the **About** dialog.

Visual Studio can install the latest .NET Core SDK and runtime.

Select a workload

When installing or modifying Visual Studio, select one or more of the following workloads, depending on the kind of application you're building:

- The **.NET Core cross-platform development** workload in the **Other Toolsets** section.
- The **ASP.NET and web development** workload in the **Web & Cloud** section.
- The **Azure development** workload in the **Web & Cloud** section.
- The **.NET desktop development** workload in the **Desktop & Mobile** section.

Install alongside Visual Studio Code

Visual Studio Code is a powerful and lightweight source code editor that runs on your desktop. Visual Studio Code is available for Windows, macOS, and Linux.

While Visual Studio Code doesn't come with an automated .NET Core installer like Visual Studio does, adding .NET Core support is simple.

1. Download and install Visual Studio Code.
2. Download and install the .NET Core SDK.
3. Install the C# extension from the Visual Studio Code marketplace.

Download and manually install

As an alternative to the Windows installers for .NET Core, you can download and manually install the SDK or runtime. Manual install is usually performed as part of continuous integration testing. For a developer or user, it's generally better to use an installer.

Both .NET Core SDK and .NET Core Runtime can be manually installed after they've been downloaded. If you install .NET Core SDK, you don't need to install the corresponding runtime. First, download a binary release for either the SDK or the runtime from one of the following sites:

- ✓◻ .NET 5.0 preview downloads
- ✓◻ .NET Core 3.1 downloads
- ✓◻ .NET Core 2.1 downloads

Create a directory to extract .NET to, for example %USERPROFILE%dotnet. Then, extract the downloaded zip file into that directory.

By default, .NET Core CLI commands and apps won't use .NET Core installed in this way and you must explicitly choose to use it. To do so, change the environment variables with which an application is started:

This approach lets you install multiple versions into separate locations, then explicitly choose which install location an application should use by running the application with environment variables pointing at that location.

When DOTNET_MULTILEVEL_LOOKUP is set to 0, .NET Core ignores any globally installed .NET Core version. Remove that environment setting to let .NET Core consider the default global install location when selecting the best framework for running the application. The default is typically C:Program Filesdotnet, which is where the installers install .NET Core.

Visual Studio 2015 For Mac Free Download

Docker

Visual Studio 2012 Free Edition

Containers provide a lightweight way to isolate your application from the rest of the host system. Containers on the same machine share just the kernel and use resources given to your application.

.NET Core can run in a Docker container. Official .NET Core Docker images are published to the Microsoft Container Registry (MCR) and are discoverable at the Microsoft .NET Core Docker Hub repository. Each repository contains images for different combinations of the .NET (SDK or Runtime) and OS that you can use.

Microsoft provides images that are tailored for specific scenarios. For example, the ASP.NET Core repository provides images that are built for running ASP.NET Core apps in production.

For more information about using .NET Core in a Docker container, see Introduction to .NET and Docker and Samples.

Next steps

- How to check if .NET Core is already installed.
- Tutorial: Hello World tutorial.
- Tutorial: Create a new app with Visual Studio Code.
- Tutorial: Containerize a .NET Core app.