**WHAT IS .NET-FRAMEWORK**

.NET Framework is a managed execution environment for Windows that provides a variety of services to its running apps. It consists of two major components: the common language runtime (CLR), which is the execution engine that handles running apps, and the .NET Framework Class Library, which provides a library of tested, reusable code that developers can call from their own apps. The services that .NET Framework provides to running apps include the following:

* Memory management. In many programming languages, programmers are responsible for allocating and releasing memory and for handling object lifetimes. In .NET Framework apps, the CLR provides these services on behalf of the app.
* A common type system. In traditional programming languages, basic types are defined by the compiler, which complicates cross-language interoperability. In .NET Framework, basic types are defined by the .NET Framework type system and are common to all languages that target .NET Framework.
* An extensive class library. Instead of having to write vast amounts of code to handle common low-level programming operations, programmers use a readily accessible library of types and their members from the .NET Framework Class Library.
* Development frameworks and technologies. .NET Framework includes libraries for specific areas of app development, such as ASP.NET for web apps, ADO.NET for data access, Windows Communication Foundation for service-oriented apps, and Windows Presentation Foundation for Windows desktop apps.
* Language interoperability. Language compilers that target .NET Framework emit an intermediate code named Common Intermediate Language (CIL), which, in turn, is compiled at run time by the common language runtime. With this feature, routines written in one language are accessible to other languages, and programmers focus on creating apps in their preferred languages.
* Version compatibility. With rare exceptions, apps that are developed by using a particular version of .NET Framework run without modification on a later version.
* Side-by-side execution. .NET Framework helps resolve version conflicts by allowing multiple versions of the common language runtime to exist on the same computer. This means that multiple versions of apps can coexist and that an app can run on the version of .NET Framework with which it was built. Side-by-side execution applies to the .NET Framework version groups 1.0/1.1, 2.0/3.0/3.5, and 4/4.5.x/4.6.x/4.7.x/4.8.x.
* Multitargeting. By targeting [.NET Standard](https://learn.microsoft.com/en-us/dotnet/standard/net-standard), developers create class libraries that work on multiple .NET Framework platforms supported by that version of the standard. For example, libraries that target .NET Standard 2.0 can be used by apps that target .NET Framework 4.6.1, .NET Core 2.0, and UWP 10.0.16299.

## **NET Framework for users**

If you don't develop .NET Framework apps, but you use them, you aren't required to have specific knowledge about .NET Framework or its operation. For the most part, the framework is completely transparent to users.

If you're using the Windows operating system, .NET Framework may already be installed on your computer. In addition, if you install an app that requires .NET Framework, the app's setup program might install a specific version of the framework on your computer. In some cases, you may see a dialog box that asks you to install .NET Framework. If you've just tried to run an app when this dialog box appears and if your computer has internet access, you can go to a webpage that lets you install the missing version of .NET Framework. For more information, see the [Installation guide](https://learn.microsoft.com/en-us/dotnet/framework/install/).

In general, you shouldn't uninstall versions of .NET Framework that are installed on your computer. There are two reasons for this:

* If an app that you use depends on a specific version of .NET Framework, that app may break if that version is removed.
* Some versions of .NET Framework are in-place updates to earlier versions. For example, .NET Framework 3.5 is an in-place update to version 2.0, and .NET Framework 4.8 is an in-place update to versions 4 through 4.7.2. For more information, see [.NET Framework Versions and Dependencies](https://learn.microsoft.com/en-us/dotnet/framework/migration-guide/versions-and-dependencies).

On Windows versions before Windows 8, if you do choose to remove .NET Framework, always use **Programs and Features** from Control Panel to uninstall it. Never remove a version of .NET Framework manually. On Windows 8 and above, .NET Framework is an operating system component and cannot be independently uninstalled.

Multiple versions of .NET Framework can coexist on a single computer at the same time. This means that you don't have to uninstall previous versions in order to install a later version.

## **.NET Framework for developers**

If you're a developer, choose any programming language that supports .NET Framework to create your apps. Because .NET Framework provides language independence and interoperability, you interact with other .NET Framework apps and components regardless of the language with which they were developed.

To develop .NET Framework apps or components, do the following:

1. If it's not preinstalled on your operating system, install the version of .NET Framework that your app will target. The current versions are .NET Framework 4.8 and .NET Framework 4.8.1. .NET Framework 4.8.1 is [available for download](https://learn.microsoft.com/en-us/dotnet/framework/install/guide-for-developers) on the latest versions of Windows and Windows Server. .NET Framework 4.8 is preinstalled on Windows 10 May 2019 Update, Windows 10 November 2019 Update, Windows 10 May 2020 Update, and Windows 10 October 2020 Update, and it's available for download on earlier versions of the Windows operating system. For .NET Framework system requirements, see [System Requirements](https://learn.microsoft.com/en-us/dotnet/framework/get-started/system-requirements). For information on installing other versions of .NET Framework, see [Installation Guide](https://learn.microsoft.com/en-us/dotnet/framework/install/guide-for-developers). Additional .NET Framework packages are released out of band, which means that they're released on a rolling basis outside of any regular or scheduled release cycle. For information about these packages, see [.NET Framework and Out-of-Band Releases](https://learn.microsoft.com/en-us/dotnet/framework/get-started/the-net-framework-and-out-of-band-releases).
2. Select the language or languages supported by the .NET Framework version that you intend to use to develop your apps. A number of languages are available, including [Visual Basic](https://learn.microsoft.com/en-us/dotnet/visual-basic/), [C#](https://learn.microsoft.com/en-us/dotnet/csharp/), [F#](https://learn.microsoft.com/en-us/dotnet/fsharp/), and [C++/CLI](https://learn.microsoft.com/en-us/cpp/dotnet/dotnet-programming-with-cpp-cli-visual-cpp) from Microsoft. (A programming language that allows you to develop apps for .NET Framework adheres to the [Common Language Infrastructure (CLI) specification](https://visualstudio.microsoft.com/license-terms/ecma-c-common-language-infrastructure-standards/).)
3. Select and install the development environment to use to create your apps and that supports your selected programming language or languages. The Microsoft integrated development environment (IDE) for .NET Framework apps is [Visual Studio](https://visualstudio.microsoft.com/vs/?utm_medium=microsoft&utm_source=docs.microsoft.com&utm_campaign=inline+link). It's available in a number of editions

**HISTORY OF .NET-FRAMEWORK**

Microsoft started development on the [.NET Framework](https://en.wikipedia.org/wiki/.NET_Framework) in the late 1990s originally under the name of Next Generation Windows Services (NGWS). By late 2001 the first beta versions of .NET 1.0 were released.[[1]](https://en.wikipedia.org/wiki/.NET_Framework_version_history#cite_note-1) The first version of .NET Framework was released on 13 February 2002, bringing [managed code](https://en.wikipedia.org/wiki/Managed_code) to [Windows NT 4.0](https://en.wikipedia.org/wiki/Windows_NT_4.0), [98](https://en.wikipedia.org/wiki/Windows_98), [2000](https://en.wikipedia.org/wiki/Windows_2000), [ME](https://en.wikipedia.org/wiki/Windows_ME) and [XP](https://en.wikipedia.org/wiki/Windows_XP).

Since the first version, Microsoft has released nine more upgrades for .NET Framework, seven of which have been released along with a new version of [Visual Studio](https://en.wikipedia.org/wiki/Visual_Studio). Two of these upgrades, .NET Framework 2.0 and 4.0, have upgraded [Common Language Runtime](https://en.wikipedia.org/wiki/Common_Language_Runtime) (CLR). New versions of .NET Framework replace older versions when the CLR version is the same.

The .NET Framework family also includes two versions for [mobile](https://en.wikipedia.org/wiki/Mobile_computing) or [embedded device](https://en.wikipedia.org/wiki/Embedded_device) use. A reduced version of the framework, the [.NET Compact Framework](https://en.wikipedia.org/wiki/.NET_Compact_Framework), is available on [Windows CE](https://en.wikipedia.org/wiki/Windows_CE) platforms, including [Windows Mobile](https://en.wikipedia.org/wiki/Windows_Mobile) devices such as [smartphones](https://en.wikipedia.org/wiki/Smartphone). Additionally, the [.NET Micro Framework](https://en.wikipedia.org/wiki/.NET_Micro_Framework) is targeted at severely resource-constrained devices.

.NET Framework 4.8 was the final version of .NET Framework, future work going into the rewritten and [cross-platform](https://en.wikipedia.org/wiki/Cross-platform_software) [.NET Core](https://en.wikipedia.org/wiki/.NET_Core) platform, which shipped as [.NET 5](https://en.wikipedia.org/wiki/.NET_5) in November 2020