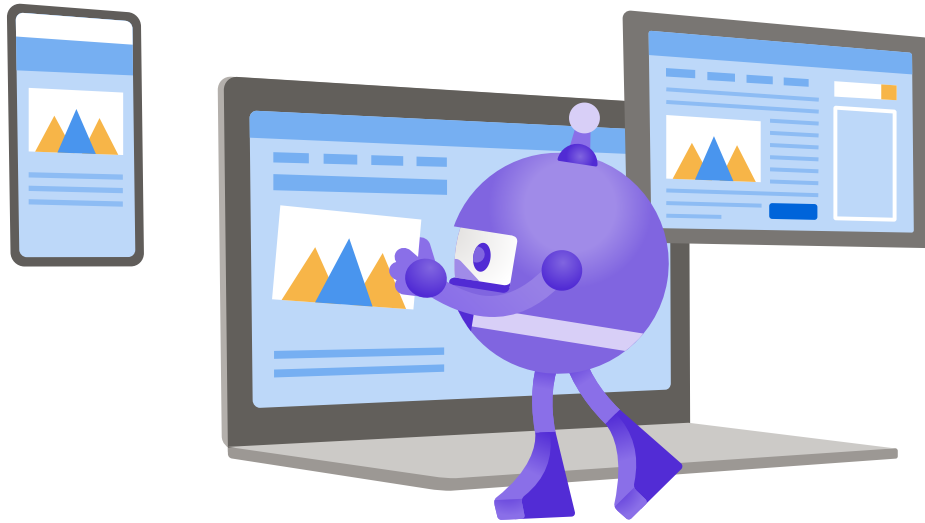


What is .NET Framework? A software development framework | .NET

Archived: 2026-04-05 13:49:44 UTC



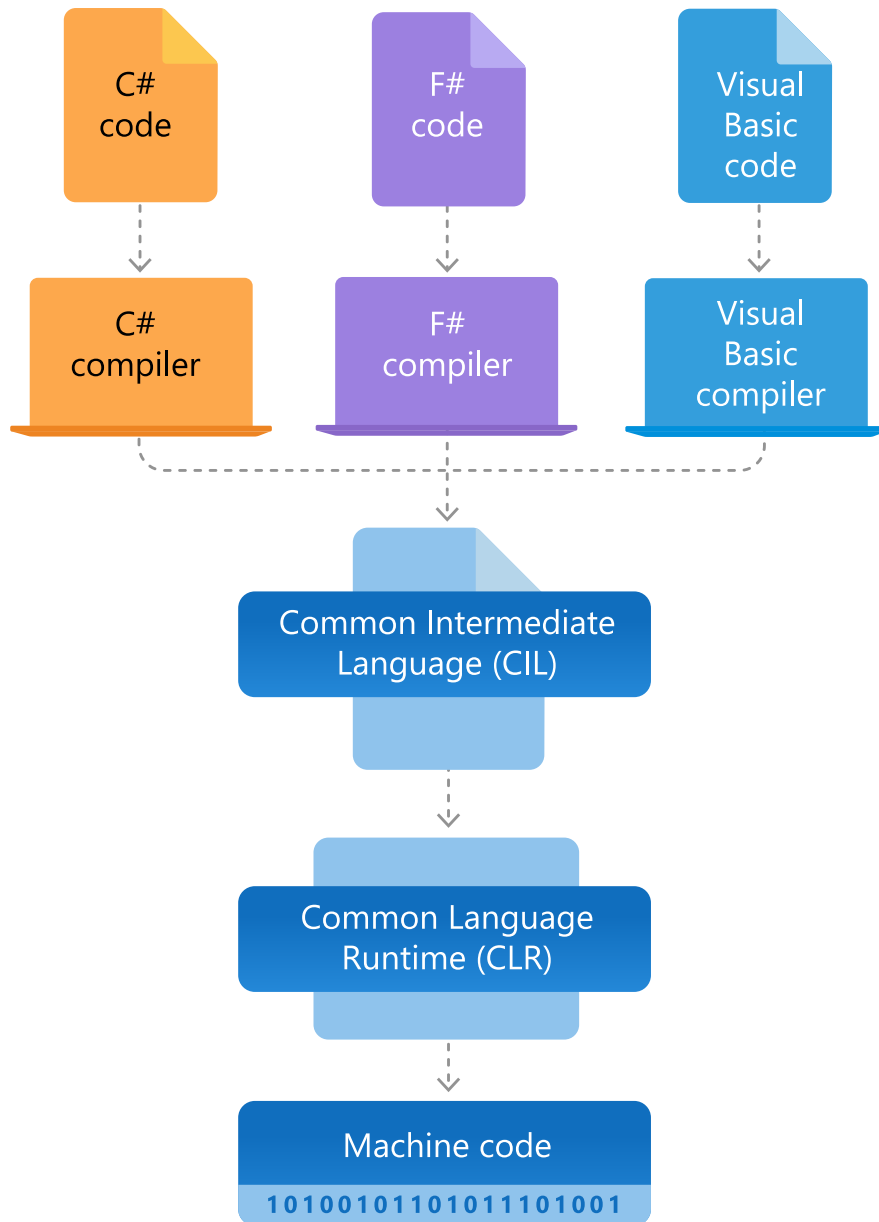
.NET and .NET Framework

[.NET is a developer platform](#) made up of [tools](#), [programming languages](#), and libraries for building many different types of applications.

There are various implementations of .NET. Each implementation allows .NET code to execute in different places—Linux, macOS, Windows, iOS, Android, and many more.

1. **.NET Framework** is the original implementation of .NET. It supports running websites, services, desktop apps, and more on Windows.
2. **.NET** is a cross-platform implementation for running websites, services, and console apps on Windows, Linux, and macOS. [.NET is open source](#) on GitHub. .NET was previously called .NET Core.
3. [Xamarin/Mono](#) is a .NET implementation for running apps on all the major mobile operating systems, including iOS and Android.

[.NET Standard](#) is a formal specification of the APIs that are common across .NET implementations. This allows the same code and libraries to run on different implementations.



Architecture of .NET Framework

The two major components of .NET Framework are the Common Language Runtime and the .NET Framework Class Library.

- The **Common Language Runtime (CLR)** is the execution engine that handles running applications. It provides services like thread management, garbage collection, type-safety, exception handling, and more.
- The **Class Library** provides a set of APIs and types for common functionality. It provides types for strings, dates, numbers, etc. The Class Library includes APIs for reading and writing files, connecting to databases, drawing, and more.

.NET applications are written in the C#, F#, or Visual Basic programming language. Code is compiled into a language-agnostic Common Intermediate Language (CIL). Compiled code is stored in assemblies—files with a

.dll or .exe file extension.

When an app runs, the CLR takes the assembly and uses a just-in-time compiler (JIT) to turn it into machine code that can execute on the specific architecture of the computer it is running on.

[Read more: What is the .NET Framework](#)

Release history

The following releases have reached end of life, meaning they're no longer supported. We recommend moving to a supported release.

Out of support versions

Version	Release date	End of support
.NET Framework 4.6.1	November 30, 2015	April 26, 2022
.NET Framework 4.6	July 20, 2015	April 26, 2022
.NET Framework 4.5.2	May 5, 2014	April 26, 2022
.NET Framework 4.5.1	October 17, 2013	January 12, 2016
.NET Framework 4.5	August 15, 2012	January 12, 2016
.NET Framework 4.0	April 12, 2010	January 12, 2016

.NET Framework FAQ

What is .NET Framework used for?

.NET Framework is used to create and run software applications. .NET apps can run on many operating systems, using different implementations of .NET. .NET Framework is used for running .NET apps on Windows.

Who uses .NET Framework?

Software developers and the users of their applications both use .NET Framework:

- Users of applications built with the .NET Framework need to have .NET Framework installed. In most cases, .NET Framework is already installed with Windows. If needed, you can [download .NET Framework](#).
- Software developers use .NET Framework to build many different types of applications—websites, services, desktop apps, and more with Visual Studio. Visual Studio is an integrated development environment (IDE) that provides development productivity tools and debugging capabilities. See the [.NET customer showcase](#) for examples of what people are building with .NET.

Why do I need .NET Framework?

You need .NET Framework installed to run applications on Windows that were created using .NET Framework. It's already included in many versions of Windows. You only need to [download and install .NET Framework](#) if prompted to do so.

How does .NET Framework work?

.NET Framework applications are written in C#, F#, or Visual Basic and compiled to Common Intermediate Language (CIL). The Common Language Runtime (CLR) runs .NET applications on a given machine, converting the CIL to machine code. See [Architecture of .NET Framework](#) for more info.

What are the main components/features of .NET Framework?

The two major components of .NET Framework are the Common Language Runtime (CLR) and the .NET Framework Class Library. The CLR is the execution engine that handles running applications. The Class Library provides a set of APIs and types for common functionality. See [Architecture of .NET Framework](#) for more info.

What is the difference between .NET and .NET Framework?

.NET and .NET Framework share many of the same components and you can share code across the two. Some key differences include:

- .NET is cross-platform and runs on Linux, macOS, and Windows. .NET Framework only runs on Windows.
- .NET is open-source and accepts contributions from the community. The .NET Framework source code is available but doesn't take direct contributions.
- All of the innovation happens in .NET.
- .NET Framework is included in Windows and automatically updated machine-wide by Windows Update. .NET is shipped independently.

See [.NET vs. .NET Framework for server apps](#) for more details.

Can you have multiple .NET Frameworks installed?

Some versions of .NET Framework are installed side-by-side, while others will upgrade an existing version (known as an in-place update). In-place updates occur when two .NET Framework versions share the same CLR version.

For example, installing .NET Framework 4.8 on a machine with .NET Framework 4.7.2 and 3.5 installed will perform an in-place update of the 4.7.2 installation and leave 3.5 installed separately.

CLR version used by .NET Framework versions

.NET Framework version	CLR version
.NET Framework 4.x	4.0
.NET Framework 2.x and 3.x	2.0

.NET Framework version	CLR version
.NET Framework 1.1	1.1
.NET Framework 1.0	1.0

How much does .NET Framework cost?

.NET Framework is free, like the rest of the .NET platform. There are no fees or licensing costs, including for commercial use. See [.NET is free](#) for more details.

Which version of .NET Framework should I use?

In most cases, you should use the latest stable release. Currently, that's [.NET Framework 4.8.1](#).

Applications that were created with any 4.x version of .NET Framework will run on .NET Framework 4.8.1 . To run an application that was created for an earlier version (for example, .NET Framework 3.5), you should install that version. See [Download .NET Framework](#) for a complete list.

What is the support policy for .NET Framework?

.NET Framework 4.8.1 is the latest version of .NET Framework and will continue to be distributed with future releases of Windows. As long as it is installed on a supported version of Windows, .NET Framework 4.8.1 will continue to also be supported.

See the [.NET Framework support policy](#) for more details

Source: <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet-framework>