.NET Core Overview

.NET Core is a new version of .NET Framework, which is a free, open-source, general-purpose development platform maintained by Microsoft. It is a cross-platform framework that runs on Windows, macOS, and Linux operating systems.

.NET Core Framework can be used to build different types of applications such as mobile, desktop, web, cloud, IoT, machine learning, microservices, game, etc.

.NET Core is written from scratch to make it modular, lightweight, fast, and cross-platform Framework. It includes the core features that are required to run a basic .NET Core app. Other features are provided as NuGet packages, which you can add it in your application as needed. In this way, the .NET Core application speed up the performance, reduce the memory footprint and becomes easy to maintain.

Why .NET Core?

There are some limitations with the .NET Framework. For example, it only runs on the Windows platform. Also, you need to use different .NET APIs for different Windows devices such as Windows Desktop, Windows Store, Windows Phone, and Web applications. In addition to this, the .NET Framework is a machine-wide framework. Any changes made to it affect all applications taking a dependency on it. Learn more about the motivation behind .NET Core [here](https://devblogs.microsoft.com/dotnet/introducing-net-core/).

Today, it's common to have an application that runs across devices; a backend on the web server, admin front-end on windows desktop, web, and mobile apps for consumers. So, there is a need for a single framework that works everywhere. So, considering this, Microsoft created .NET Core. The main objective of .NET Core is to make .NET Framework open-source, cross-platform compatible that can be used in a wide variety of verticals, from the data center to touch-based devices.

.NET Core Characteristics

**Open-source Framework:** .NET Core is an [open-source framework](https://dotnet.microsoft.com/platform/open-source) maintained by Microsoft and available on GitHub under [MIT](https://github.com/dotnet/runtime/blob/master/LICENSE.TXT) and [Apache 2](https://www.apache.org/licenses/LICENSE-2.0) licenses. It is a [.NET Foundation project](https://dotnetfoundation.org/).

You can view, download, or contribute to the source code using the following GitHub repositories:

* Language compiler platform Roslyn: <https://github.com/dotnet/roslyn>
* .NET Core runtime: <https://github.com/dotnet/runtime>
* .NET Core SDK repository. <https://github.com/dotnet/sdk>
* ASP.NET Core repository. <https://github.com/dotnet/aspnetcore>

**Cross-platform:** .NET Core runs on Windows, macOS, and Linux operating systems. There are different runtime for each operating system that executes the code and generates the same output.

**Consistent across Architectures:** Execute the code with the same behavior in different instruction set architectures, including x64, x86, and ARM.

**Wide-range of Applications:** Various types of applications can be developed and run on .NET Core platform such as mobile, desktop, web, cloud, IoT, machine learning, microservices, game, etc.

**Supports Multiple Languages:** You can use C#, F#, and Visual Basic programming languages to develop .NET Core applications. You can use your favorite IDE, including Visual Studio 2017/2019, Visual Studio Code, Sublime Text, Vim, etc.

**Modular Architecture:** .NET Core supports modular architecture approach using NuGet packages. There are different NuGet packages for various features that can be added to the .NET Core project as needed. Even the .NET Core library is provided as a NuGet package. The NuGet package for the default .NET Core application model is [Microsoft.NETCore.App](https://www.nuget.org/packages/Microsoft.NETCore.App).

This way, it reduces the memory footprint, speeds up the performance, and easy to maintain.

**CLI Tools:** .NET Core includes [CLI tools](https://www.tutorialsteacher.com/core/net-core-command-line-interface) (Command-line interface) for development and continuous-integration.

**Flexible Deployment:** .NET Core application can be deployed user-wide or system-wide or with [Docker Containers](https://docs.microsoft.com/en-us/dotnet/core/docker/introduction).

**Compatibility:** Compatible with .NET Framework and Mono APIs by using [.NET Standard specification](https://docs.microsoft.com/en-us/dotnet/standard/net-standard).

.NET Core Version History

| Version | Latest Version | Visual Studio | Release Date | End of Support |
| --- | --- | --- | --- | --- |
| .NET 5 | Preview 1 | VS 2019 | 16th March, 2020 |  |
| .NET Core 3.x - latest | 3.1.3 | VS 209 | 24th March, 2020 | 12th March, 2022 |
| .NET Core 2.x | 2.1.17 | VS 2017, 2019 | 24th March, 2020 | 21st August, 2021 |
| .NET Core 1.x | 1.1.13 | VS 2017 | 14th May, 2019 | 27th May, 2019 |

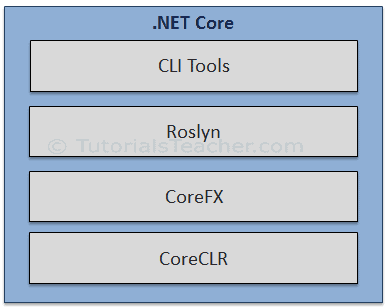
Both, .NET 3.1, and .NET Core 2.1 will have long term support.

.NET Core 3.x applications only run on .NET Core Framework.

.NET Core 2.x applications run on .NET Core as well as .NET Framework.

.NET Core Composition

The .NET Core Framework composed of the following parts:

[](https://www.tutorialsteacher.com/Content/images/core/dotnet-core-components.png).NET Core

* CLI Tools: A set of tooling for development and deployment.
* Roslyn: Language compiler for C# and Visual Basic
* CoreFX: Set of framework libraries.
* CoreCLR: A JIT based CLR (Command Language Runtime).

Useful Links

* Find .NET Core type and member declarations, files, and assemblies on [.NET Core Source Browser](https://source.dot.net/).
* Get the help on .NET Core types on [.NET Core API Reference Docs](https://docs.microsoft.com/dotnet/api/?view=netcore-3.0).
* Learn about .NET Core in detail on [https://docs.microsoft.com/en-us/dotnet/core](https://docs.microsoft.com/en-us/dotnet/core/)
* Download .NET Core runtime and SDK from <https://dotnet.microsoft.com/download/dotnet-core>