Confusion About .NET vs .NET Core vs .NET Standard vs .NET Framework

Asked 1 year ago Modified 5 months ago Viewed 27k times

39

I'm kind of new to .NET area.

There is big confusion about all these which I really couldn't figure out. I searched a lot but couldn't find any simple and straight forward explanation for .NET (also by mean all other different names of versions like in title as well)

So my question is why are there different names of .NET(as I wrote in the title). Which one is for who?

c# .net .net-core .net-standard .net-framework-version

Share Edit Follow Flag

edited Dec 22, 2023 at 8:34

Guru Stron
134k 11 146 183

asked Jul 23, 2023 at 12:48
livan3li
660 1 7 18

Please avoid asking multiple questions in one post. It makes it very hard to answer – Zohar Peled Jul 23, 2023 at 12:54

Alright, you are right. I'm gonna remove some of them which is not much related. – livan3li Jul 23, 2023 at 12:57

8 .NET 5 and later are just the evolution of .NET Core. They dropped the suffix because .NET Framework is no longer being developed so there was nothing to differentiate it from. .NET Standard was a way to write code that would work with Framework and Core. With no new Framework, there's no need for new Standard either. – jmcilhinney Jul 23, 2023 at 13:10

learn.microsoft.com/en-us/lifecycle/products/... – Hans Passant Jul 23, 2023 at 13:13

2 Answers

Sorted by: Highest score (default)

35

Here is the simplified explanation of .NET Standard, including the different names and versions of .NET:

.NET Standard: This is a specification that defines a common set of APIs that can be used by .NET applications. This makes it possible to write code that can run on different .NET platforms, such as .NET Framework, .NET Core, and .NET 5 and beyond.

.NET Framework: This is the original version of the .NET platform, which is mostly used for building Windows desktops and web applications. It is still supported, but it is considered a legacy technology.

.NET Core: This is a cross-platform version of .NET that was introduced to overcome some limitations of the .NET Framework. It is lightweight, fast, and suitable for building cross-platform applications that can run on Windows, macOS, and Linux. .NET Core is the recommended choice for new projects.

.NET 5 and beyond Microsoft decided to simplify the naming and converge .NET Core and .NET Framework into a single platform called .NET 5. This new platform is fully supported and is the recommended choice for new projects.

In summary, if you are starting a new project, you should use the latest version of .NET Standard, such as .NET Standard 2.1 or later. This will give you the most modern features and improvements, and it will also make your application more cross-platform.

Share Edit Follow Flag

answered Jul 23, 2023 at 13:41

Vivek Nuna
29k 31 138 240

12 You wrote this in Jun 2023: "if you are starting a new project, you should use the latest version of .NET Standard," In 2023 I would be using .NET 6 or .NET 7. (unless I have specific needs to stay with .NET Standard!). – joedotnot Jan 13 at 9:00

3 @Vivek, thanks for taking a stab at this, but I find your answer still confusing. You state that as of 2023 new projects should use .NET Standard - and I am still confused about the diff between Standard and "5 and beyond" - I don't think "5 and beyond" is an official designation. AFAIK .NET 8 (the current release) **is** .NET Core 8, and Standard is also legacy; so all new projects should be .NET Core . Can you provide feedback on that thought? – Jay Imerman Mar 28 at 14:26

1 Also, for .NET Core, every EVEN release is Long-Term-Support (LTS), while every odd release is considered "developmental" for new features. For any non-explorational/experimental code development, I would stick to even releases. If you are investigating a possible bug fix or new feature, odd releases are good. Thoughts? – Jay Imerman Mar 28 at 14:28

.NET is a cross language set of compilers and tools that are supported by an underlying set of runtime libraries. C# is the major .NET language, but also VB.NET and F# are also .NET languages.

15

.NET Framework is Windows only. The final version (v4.8.x) it is not actively being developed, but that doesn't mean it won't get security updates and it will be supported for as long as Window 11 is supported (and may be later OSs). https://dotnet.microsoft.com/en-us/platform/support/policy/dotnet-framework

.NET Core is supported on multiple operating systems (including Windows, Linux, macOs, IOS, Android OS and others). It was completely rewritten from scratch and although it shares a large compatibility with .NET framework, it is very different in many places. https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet

.NET 5 and above is just .NET core version 5 and above. 5 was chosen to be higher than both .NET core 3, and .NET framework 4.8. (there was no .NET core 4.X)

.NET Standard is an in-between library that allows code to be shared between .NET Framework and .NET Core. In theory, if you compile code to .NET standard, it should be compatible with Framework and Core.

If you are looking for advice for which version of .NET to write code for (not really allowed on SO), it would generally be .NET core, unless you require one of the many Windows only APIs that are only available in .NET Framwork (although things like WCF is now on .NET core).

Share Edit Follow Flag

edited Jul 23, 2023 at 13:16

answered Jul 23, 2023 at 13:06

Neil
11.5k 3 34 63

3 You haven't written anything about .net 5 and later (which are the newest versions) – Zohar Peled Jul 23, 2023 at 13:08

4 NET5 and later is just .NET core, there's not much to say, is there? – Neil Jul 23, 2023 at 13:10

7 That's correct, but the OP is confused by the different names so a line or two indicating that .net 5 is the successor of .net core 3.1 would probably help them. – Zohar Peled Jul 23, 2023 at 13:17

1 C# (one of the .NET languages) is absolutely being updated and released constantly. C#11 was released November 2022, and C#12 is currently in preview and will be release either later this year 2023, or early 2024. C# is a language, but .NET is a set of libraries. They are independent things learn.microsoft.com/en-us/dotnet/csharp/whats-new/... – Neil Jul 23, 2023 at 13:39

3 @Neil: "C# is a language, but .NET is a set of libraries. They are independent things" - well, not as much as you'd think. Typically, each new C# version requires a (recent) minimum level of the Framework for the precise reason that the C# version maps new language constructs to types and attributes introduced only with the current version of the libraries. – O. R. Mapper Jul 23, 2023 at 13:45