

A blue parallelogram and a light green parallelogram are positioned in the upper-left corner of the slide. The blue shape is partially behind the green one. Both shapes are oriented diagonally, matching the overall geometric theme of the background.

What is .Net
Standard?



a formal specification of .NET APIs that are available on multiple .NET implementations.

The motivation behind .NET Standard was to establish greater uniformity in the .NET ecosystem.

However, .NET 5 adopts a different approach to establishing uniformity, and this new approach eliminates the need for .NET Standard in many scenarios.

.NET 5 and .NET Standard

.NET 5 - Microsoft is actively developing.

It's a single product with a uniform set of capabilities and APIs that can be used for Windows desktop apps and cross-platform console apps, cloud services, and websites.



Which .NET Standard version to target?

When choosing a .NET Standard version to target, consider this trade-off:

- The higher the version, the more APIs are available to your library's code.
- The lower the version, the more apps and libraries can use your library.

It is recommended to use the *lowest* version of .NET Standard possible.



.NET Standard versioning rules

There are two primary versioning rules:

- Additive: .NET Standard versions are logically concentric circles: higher versions incorporate all APIs from previous versions. There are no breaking changes between versions.
- Immutable: Once shipped, .NET Standard versions are frozen.



.NET Standard problems

Here are some problems with .NET Standard that help explain why .NET 5 is the better way to share code across platforms and workloads:

- Slowness to add new APIs
- Complex versioning
- Platform-unsupported exceptions at run time