Building Great Libraries with .NET Standard

Chad Green

Code PaLOUsa 2019

August 23, 2019

Who is Chad Green

- Director of Software Development at ScholarRx
- Microsoft MVP (Developer Technology)

chadgreen@chadgreen.com

chadgreen.com

ChadGreen

ChadwickEGreen

Agenda

Building Great Libraries with .NET Standard

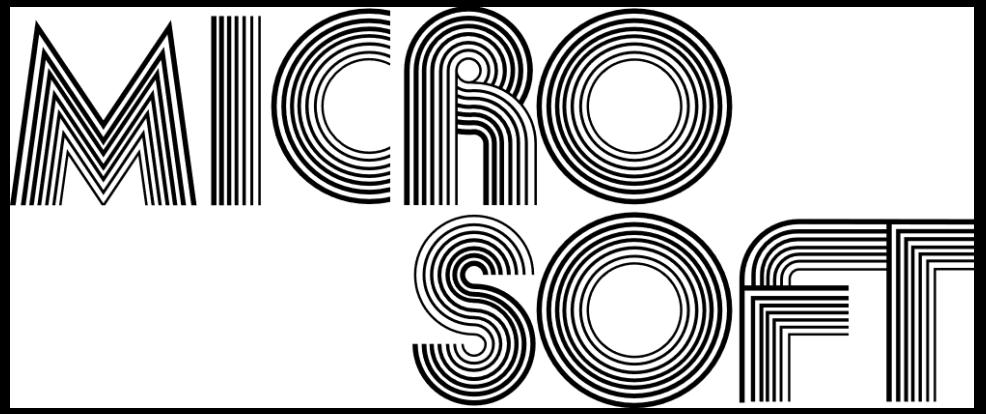
1. Quick History Lesson
2. Choosing the Right Version
3. Referencing .NET Framework Libraries
4. When to Use .NET Standard
5. Platform Specific Code
6. Guidelines for Great Libraries

Quick History Lesson

Building Great Libraries with .NET Standard

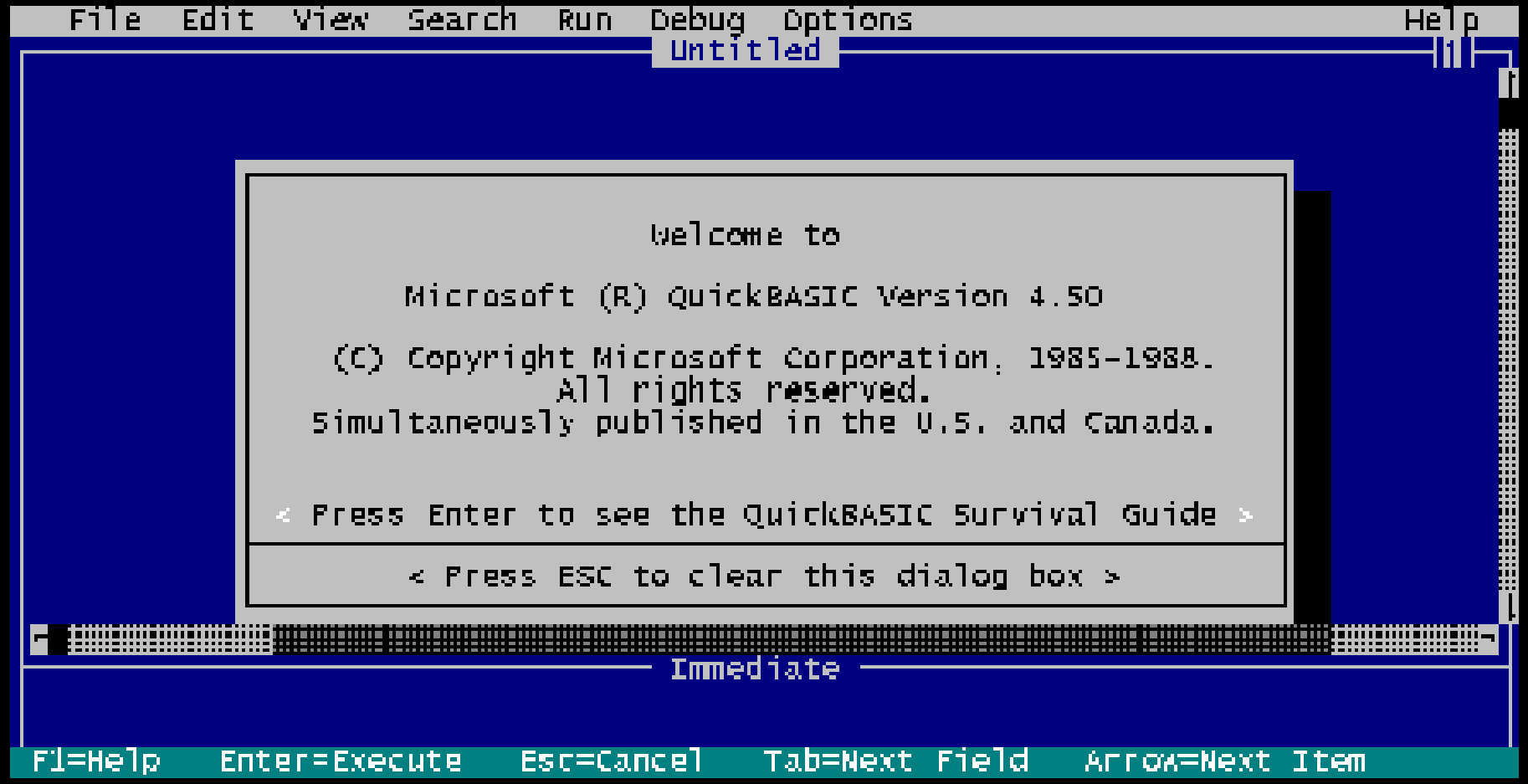
Microsoft Development Tools Up to .NET

- MicroSoft Basic



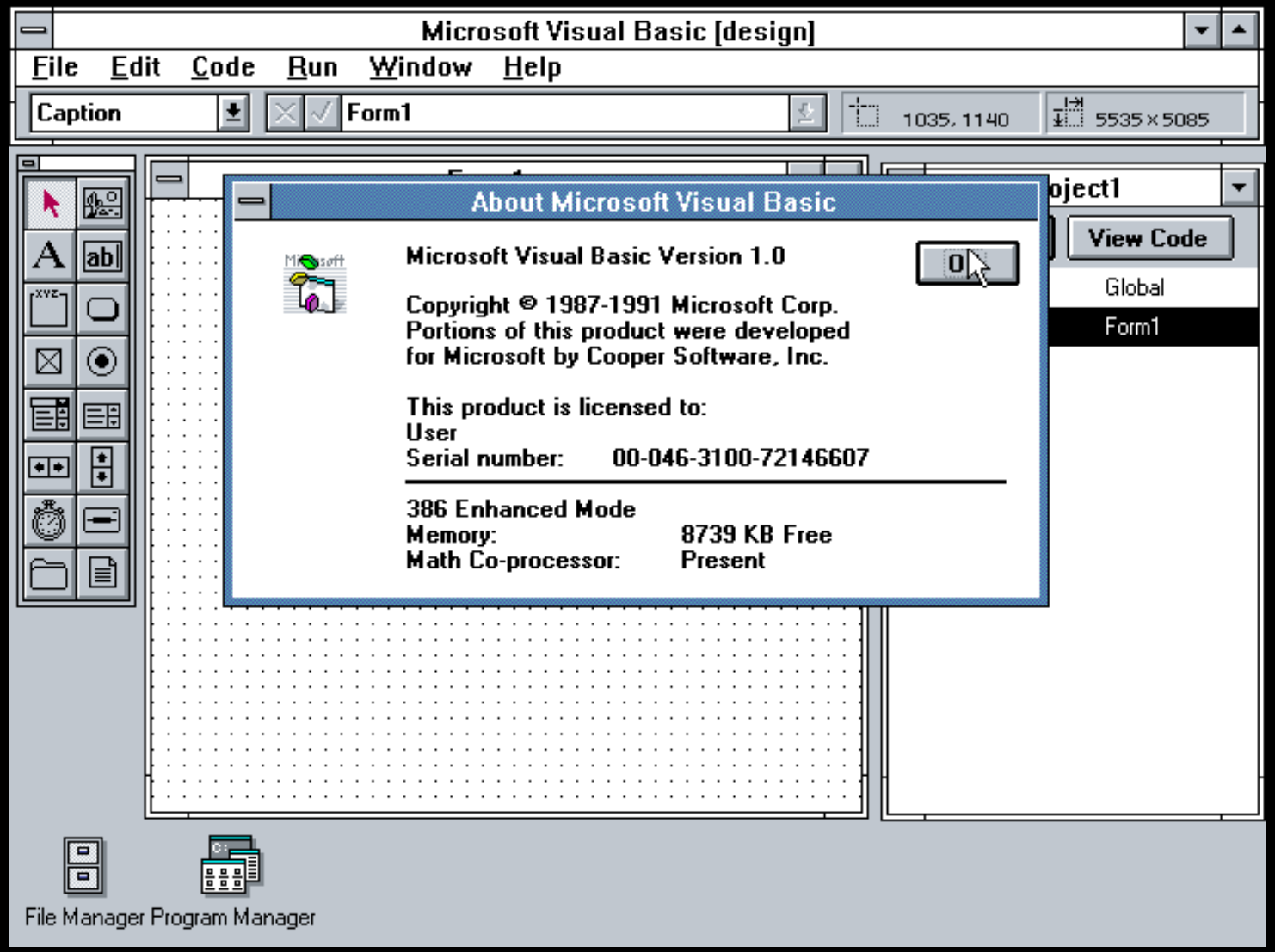
Microsoft Development Tools Up to .NET

- MicroSoft Basic
- QuickBASIC



Microsoft Development Tools Up to .NET

- MicroSoft Basic
- QuickBASIC
- Visual Basic



Introduction of Microsoft .NET

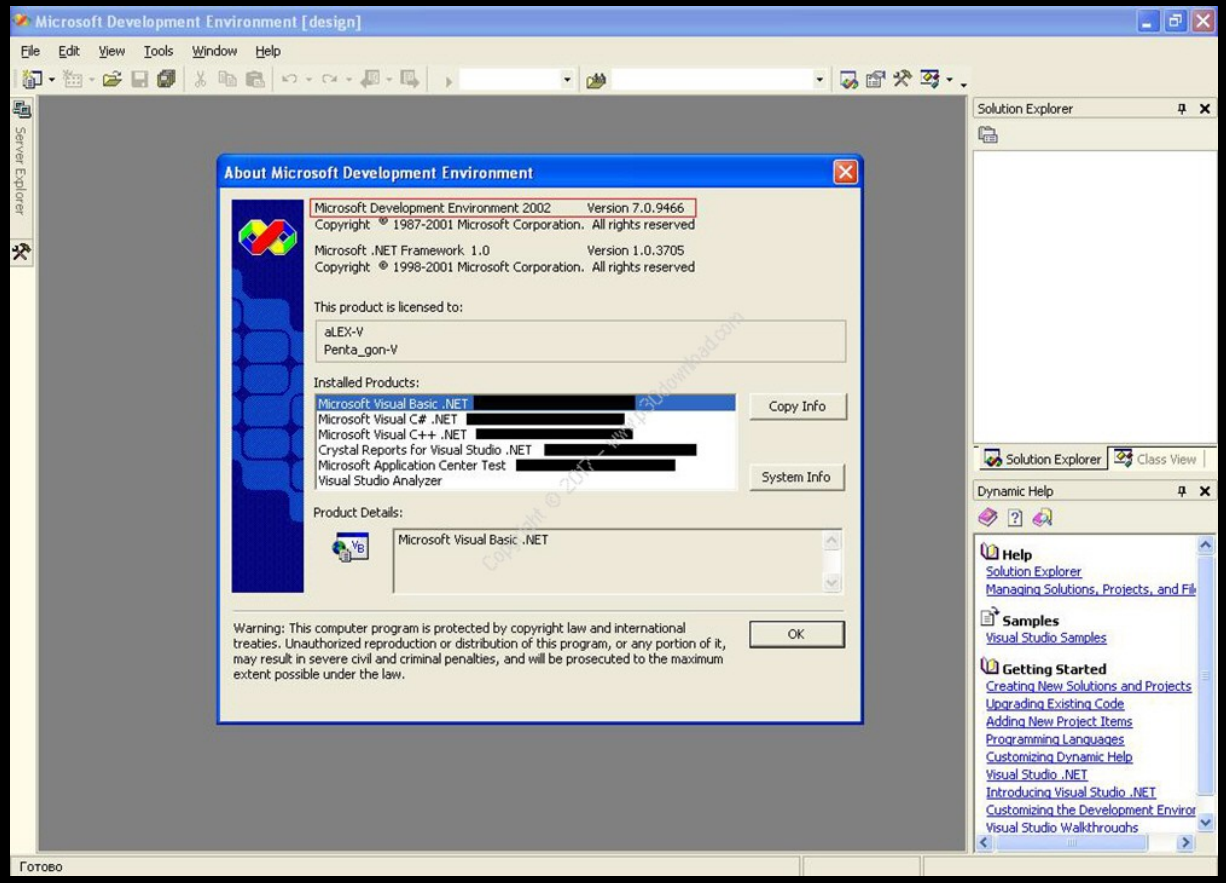
- Java
- Managed Runtime
- Runtime of many names
  - COM+
  - Compound Object Runtime (COR)
  - Universal RunTime (URT)
  - Next-Gen Windows Service (NGWS)
  - .NET

Introduction of Microsoft .NET

- Microsoft actually wanted industry support
  - Took specifications to ECMA
  - ECMA 334 – C# Language Specification
  - ECMA 335 – Common Language Infrastructure (CLI)
  - Microsoft, HP, Intel were core sponsors

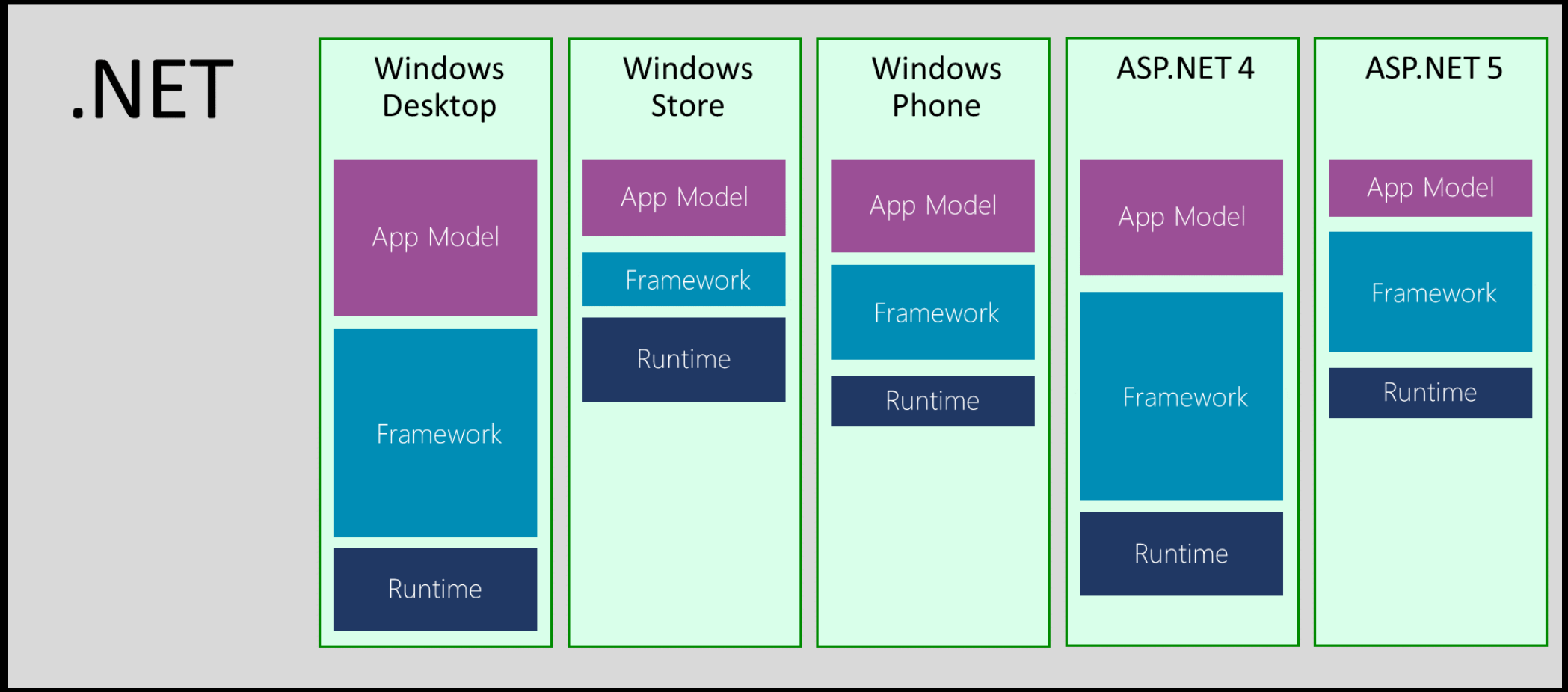
Microsoft .NET

- Introduction of CLR
- Support for object-oriented Web application development
- Use of DLL class libraries



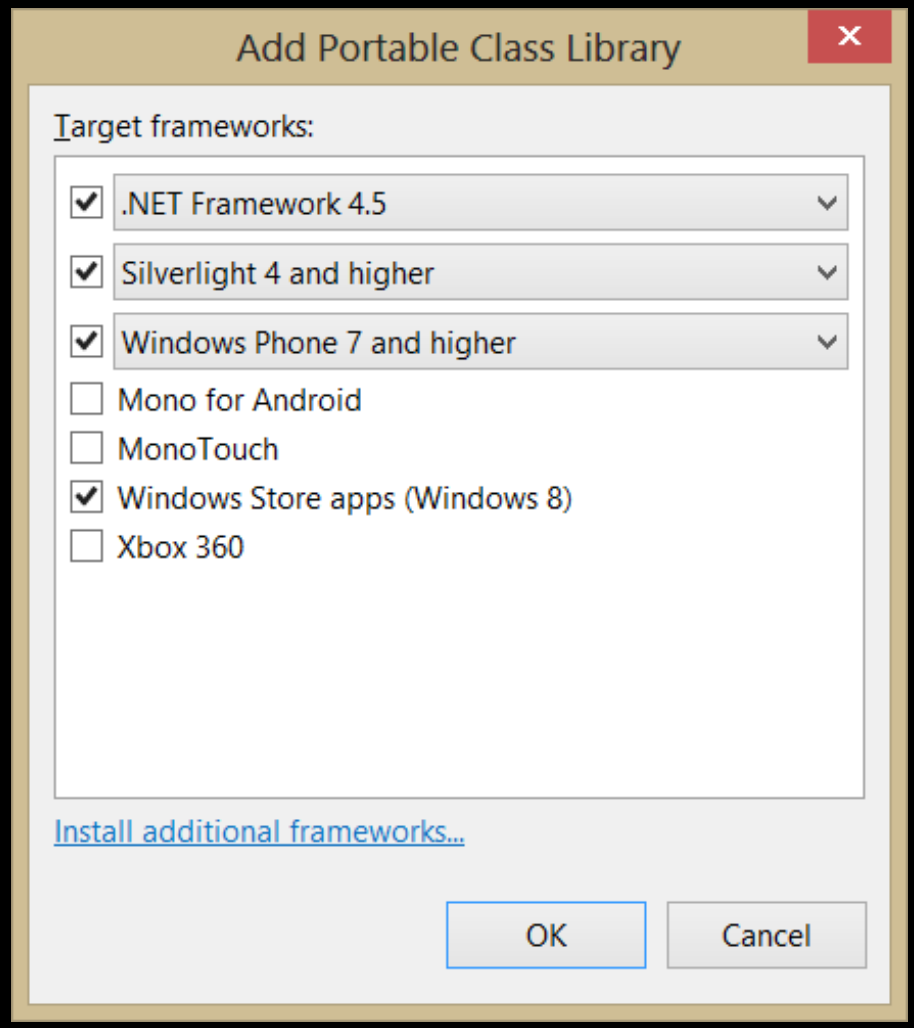
So why something different?

- .NET – a set of verticals



So why something different?

- .NET – a set of verticals
- Birth of portable class libraries



It’s a Whole New World

Building Great Libraries with .NET Standard


A whole new world

It’s a whole new world

.NET Framework	.NET Core	Xamarin
Windows Only	Windows, Linux, Mac OS	iOS, Android

It’s a whole new world

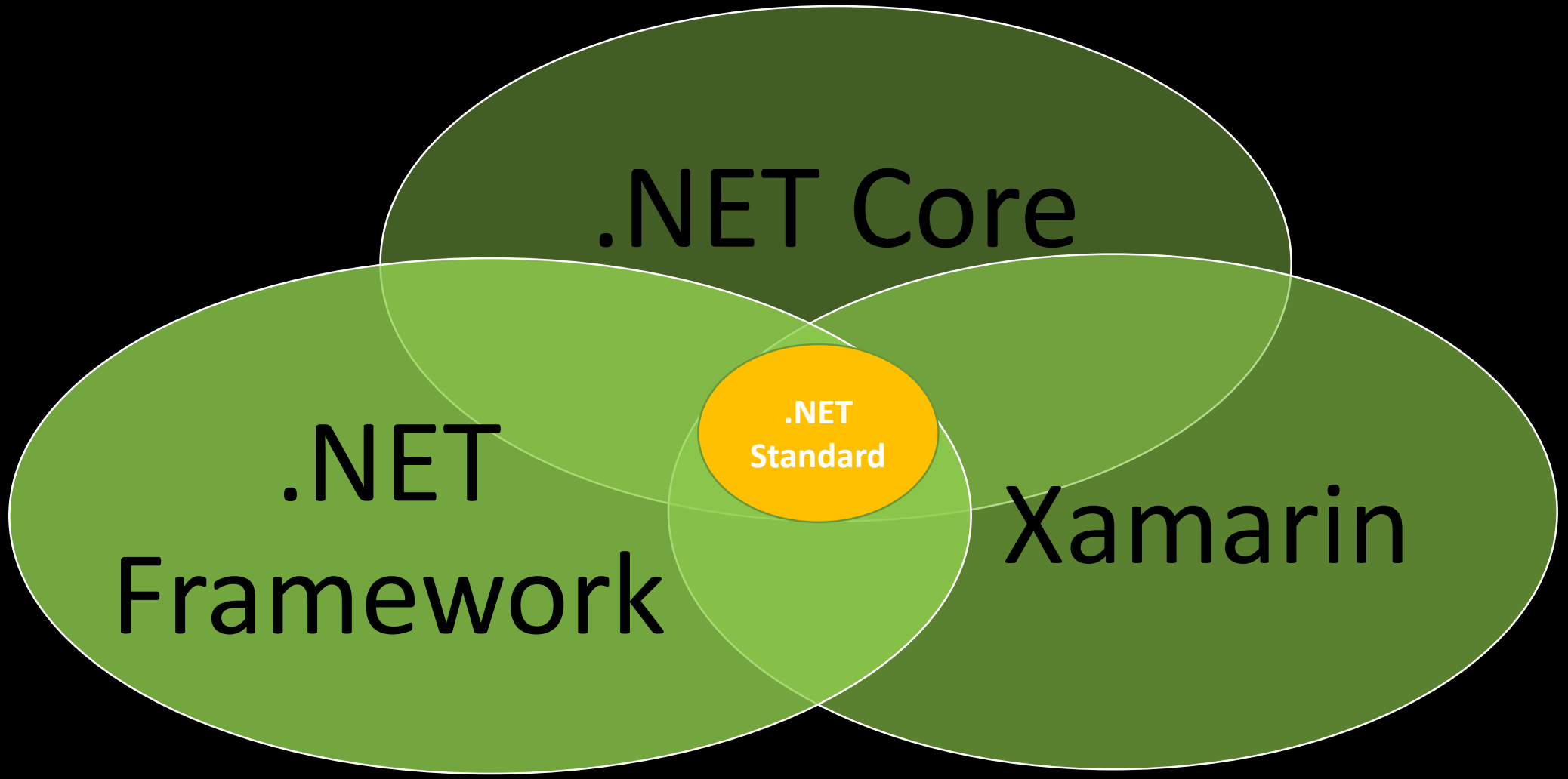
What is .NET Standard?

A Specification

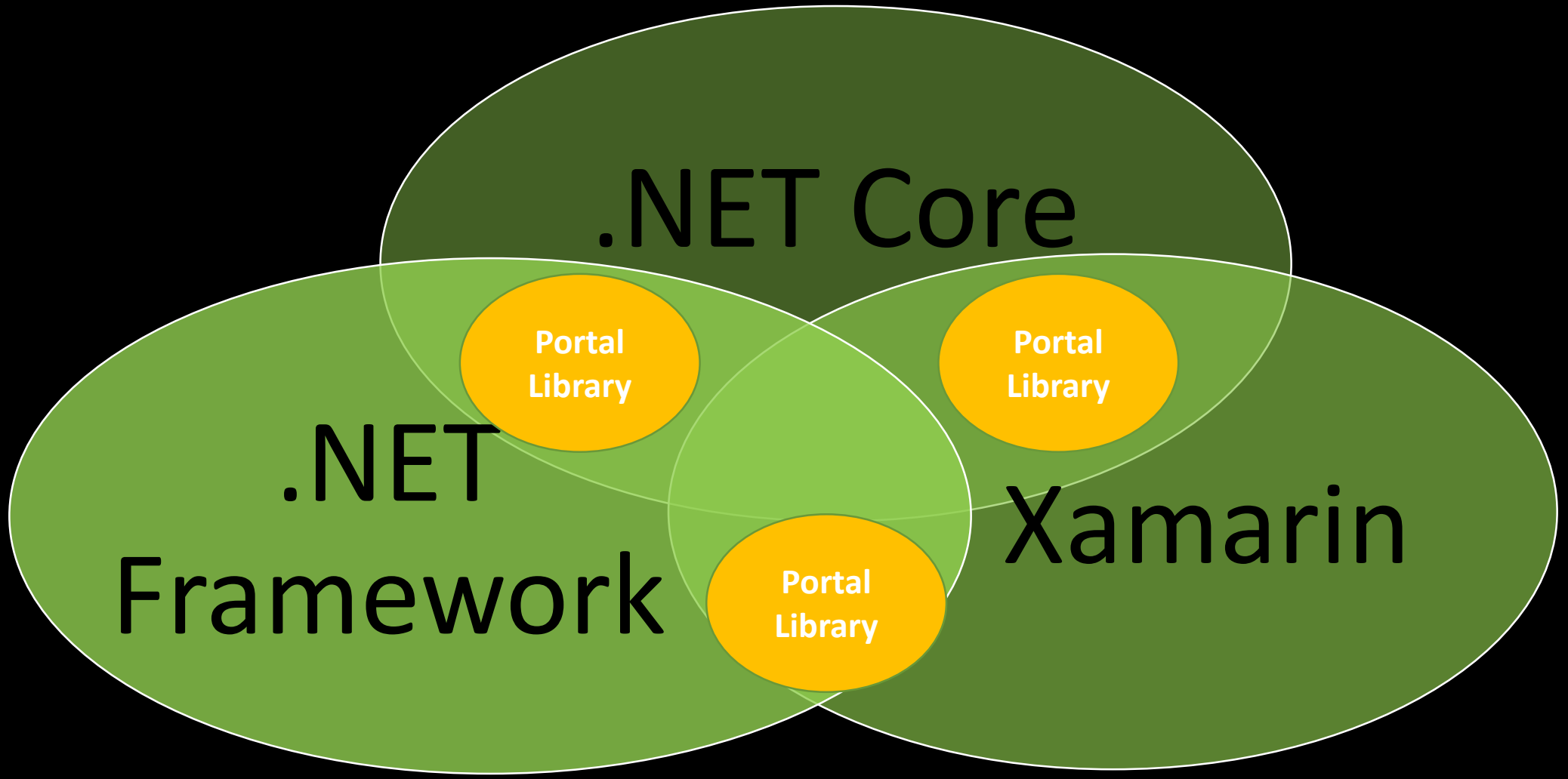
Defines a set of APIs that all .NET platforms have to implement

Not another .NET platform

What is .NET Standard?



What is .NET Standard?



What is .NET Standard?

Portable Class Libraries

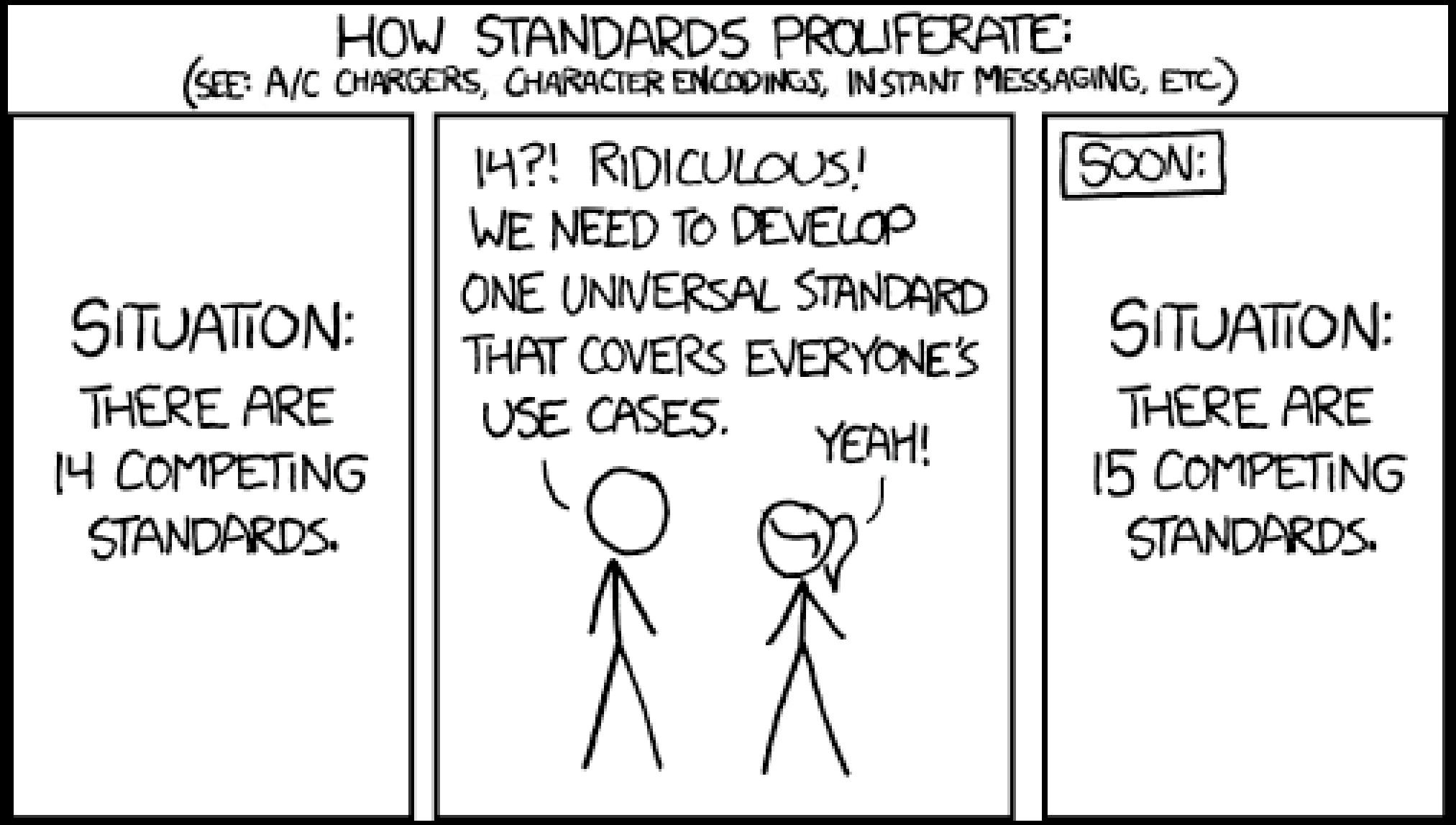
- Intersection profiles are computed
- Depend on the targeted platform
- No systematic versioning approach

Deprecated

.NET Standard

- Set of APIs that are selected by humans
- Independent from any .NET Platform
- Versioned linearly and backward compatible

What is .NET Standard?



OK, so which framework am I supposed to use?

IT Depends

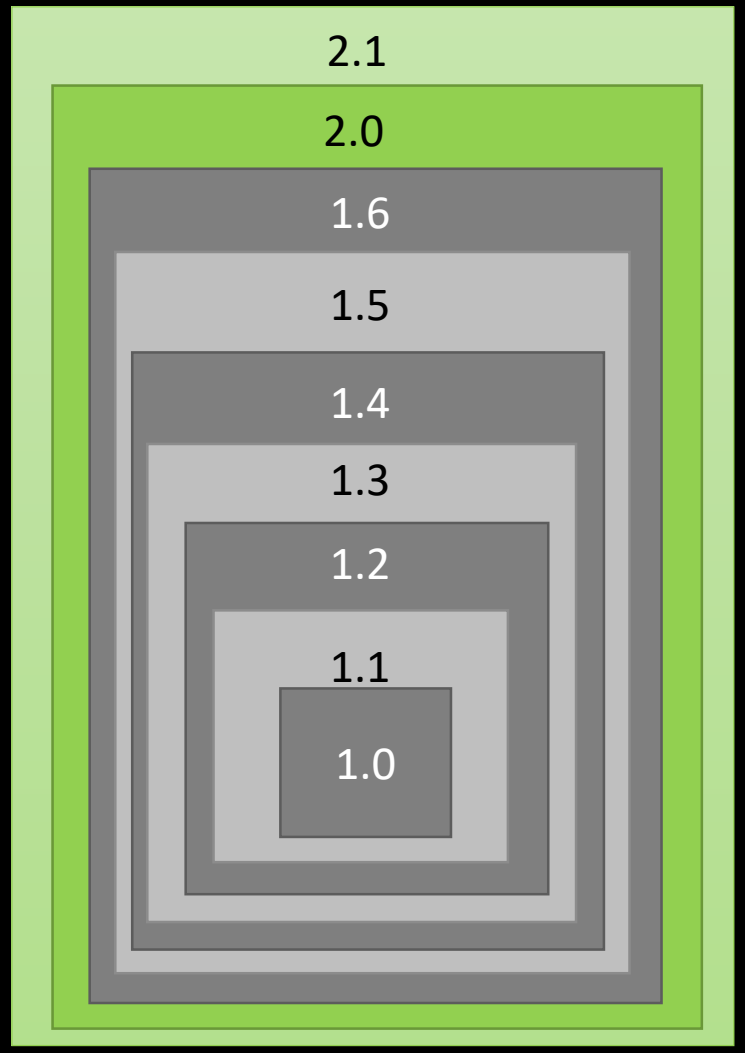
.NET Framework	.NET Core	.NET Standard
Windows Only	Windows, Linux, Mac OS	Cross .NET Platform


Demo
Building Great Libraries with .NET Standard

Choosing the Right Version

Building Great Libraries with .NET Standard

Choosing the Right Version



- A .NET Standard version contains all APIs from the previous versions
- A .NET platform implements a specific .NET Standard version
- Every .NET Standard version is immutable

Choosing the Right Version

A higher version means more APIs, but less platforms

A lower version means less APIs, but more platforms

Target the lowest possible version

Choosing the Right Version

Version: .NET Standard 1.0 ▾

Available APIs: 7,949 of 32,638

.NET Implementation	Version Support
.NET Core	✓ 1.0 ✓ 1.1 ✓ 2.0 ✓ 2.1 ✓ 2.2 ✓ 3.0
.NET Framework	✓ 4.5 ✓ 4.5.1 ✓ 4.5.2 ✓ 4.6 ✓ 4.6.1 ✓ 4.6.2 ✓ 4.7 ✓ 4.7.1 ✓ 4.7.2 ✓ 4.8
Mono	✓ 4.6 ✓ 5.4
Xamarin.iOS	✓ 10.0 ✓ 10.14
Xamarin.Android	✓ 7.0 ✓ 8.0
Universal Windows Platform	✓ 8.0 ✓ 8.1 ✓ 10.0 ✓ 10.0.16299
Unity	✓ 2018.1

Choosing the Right Version

Version: .NET Standard 1.1 ▾

Available APIs: 10,239 of 32,638

.NET Implementation	Version Support
.NET Core	✓ 1.0 ✓ 1.1 ✓ 2.0 ✓ 2.1 ✓ 2.2 ✓ 3.0
.NET Framework	✓ 4.5 ✓ 4.5.1 ✓ 4.5.2 ✓ 4.6 ✓ 4.6.1 ✓ 4.6.2 ✓ 4.7 ✓ 4.7.1 ✓ 4.7.2 ✓ 4.8
Mono	✓ 4.6 ✓ 5.4
Xamarin.iOS	✓ 10.0 ✓ 10.14
Xamarin.Android	✓ 7.0 ✓ 8.0
Universal Windows Platform	✓ 8.0 ✓ 8.1 ✓ 10.0 ✓ 10.0.16299
Unity	✓ 2018.1

Choosing the Right Version

Version: .NET Standard 1.2 ▾

Available APIs: 10,285 of 32,638

.NET Implementation	Version Support
.NET Core	✓ 1.0 ✓ 1.1 ✓ 2.0 ✓ 2.1 ✓ 2.2 ✓ 3.0
.NET Framework	× 4.5 ✓ 4.5.1 ✓ 4.5.2 ✓ 4.6 ✓ 4.6.1 ✓ 4.6.2 ✓ 4.7 ✓ 4.7.1 ✓ 4.7.2 ✓ 4.8
Mono	✓ 4.6 ✓ 5.4
Xamarin.iOS	✓ 10.0 ✓ 10.14
Xamarin.Android	✓ 7.0 ✓ 8.0
Universal Windows Platform	× 8.0 ✓ 8.1 ✓ 10.0 ✓ 10.0.16299
Unity	✓ 2018.1

Choosing the Right Version

Version: .NET Standard 1.3 ▾

Available APIs: 13,122 of 32,638

.NET Implementation	Version Support
.NET Core	✓ 1.0 ✓ 1.1 ✓ 2.0 ✓ 2.1 ✓ 2.2 ✓ 3.0
.NET Framework	× 4.5 × 4.5.1 × 4.5.2 ✓ 4.6 ✓ 4.6.1 ✓ 4.6.2 ✓ 4.7 ✓ 4.7.1 ✓ 4.7.2 ✓ 4.8
Mono	✓ 4.6 ✓ 5.4
Xamarin.iOS	✓ 10.0 ✓ 10.14
Xamarin.Android	✓ 7.0 ✓ 8.0
Universal Windows Platform	× 8.0 × 8.1 ✓ 10.0 ✓ 10.0.16299
Unity	✓ 2018.1

Choosing the Right Version

Version: .NET Standard 1.4 ▾

Available APIs: 13,140 of 32,638

.NET Implementation	Version Support
.NET Core	✓ 1.0 ✓ 1.1 ✓ 2.0 ✓ 2.1 ✓ 2.2 ✓ 3.0
.NET Framework	× 4.5 × 4.5.1 × 4.5.2 × 4.6 ✓ 4.6.1 ✓ 4.6.2 ✓ 4.7 ✓ 4.7.1 ✓ 4.7.2 ✓ 4.8
Mono	✓ 4.6 ✓ 5.4
Xamarin.iOS	✓ 10.0 ✓ 10.14
Xamarin.Android	✓ 7.0 ✓ 8.0
Universal Windows Platform	× 8.0 × 8.1 ✓ 10.0 ✓ 10.0.16299
Unity	✓ 2018.1

Choosing the Right Version

Version: .NET Standard 1.5 ▾

Available APIs: 13,355 of 32,638

.NET Implementation	Version Support
.NET Core	✓ 1.0 ✓ 1.1 ✓ 2.0 ✓ 2.1 ✓ 2.2 ✓ 3.0
.NET Framework	× 4.5 × 4.5.1 × 4.5.2 × 4.6 ✓ 4.6.1 ✓ 4.6.2 ✓ 4.7 ✓ 4.7.1 ✓ 4.7.2 ✓ 4.8
Mono	✓ 4.6 ✓ 5.4
Xamarin.iOS	✓ 10.0 ✓ 10.14
Xamarin.Android	✓ 7.0 ✓ 8.0
Universal Windows Platform	× 8.0 × 8.1 × 10.0 ✓ 10.0.16299
Unity	✓ 2018.1

Choosing the Right Version

Version: .NET Standard 1.6 ▾

Available APIs: 13,501 of 32,638

.NET Implementation	Version Support
.NET Core	✓ 1.0 ✓ 1.1 ✓ 2.0 ✓ 2.1 ✓ 2.2 ✓ 3.0
.NET Framework	× 4.5 × 4.5.1 × 4.5.2 × 4.6 ✓ 4.6.1 ✓ 4.6.2 ✓ 4.7 ✓ 4.7.1 ✓ 4.7.2 ✓ 4.8
Mono	✓ 4.6 ✓ 5.4
Xamarin.iOS	✓ 10.0 ✓ 10.14
Xamarin.Android	✓ 7.0 ✓ 8.0
Universal Windows Platform	× 8.0 × 8.1 × 10.0 ✓ 10.0.16299
Unity	✓ 2018.1

Choosing the Right Version

Version: .NET Standard 2.0 ▾

Available APIs: 32,638 of 32,638

.NET Implementation	Version Support
.NET Core	× 1.0 × 1.1 ✓ 2.0 ✓ 2.1 ✓ 2.2 ✓ 3.0
.NET Framework	× 4.5 × 4.5.1 × 4.5.2 × 4.6 ✓ 4.6.1 ✓ 4.6.2 ✓ 4.7 ✓ 4.7.1 ✓ 4.7.2 ✓ 4.8
Mono	× 4.6 ✓ 5.4
Xamarin.iOS	× 10.0 ✓ 10.14
Xamarin.Android	× 7.0 ✓ 8.0
Universal Windows Platform	× 8.0 × 8.1 × 10.0 ✓ 10.0.16299
Unity	✓ 2018.1

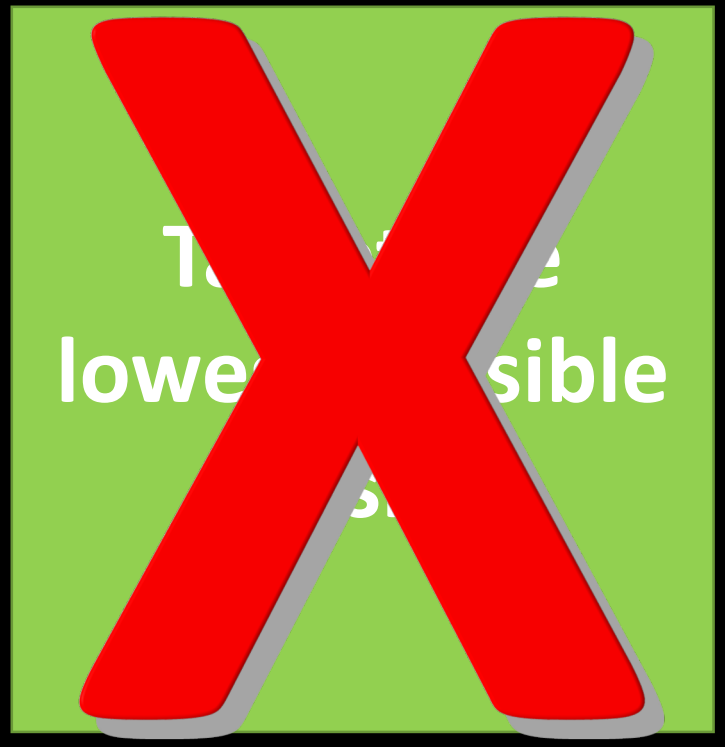
Choosing the Right Version

https://dotnet.microsoft.com/platform/dotnet-standard

Choosing the Right Version

A higher version means more APIs, but less platforms

A lower version means less APIs, but more platforms



Choosing the Right Version

.NET Standard 2.1

- Span<T>

Choosing the Right Version

.NET Standard 2.1

- Span<T>
- Foundational APIs working with spans

Choosing the Right Version

.NET Standard 2.1

- Span<T>
- Foundational APIs working with spans
- Reflection emit

Choosing the Right Version

.NET Standard 2.1

- Span<T>
- Foundational APIs working with spans
- Reflection emit
- SIMD (Single Instruction, Multiple Data)
- ValueTask and ValueTask<T>

Choosing the Right Version

.NET Standard 2.1

- Span<T>
- Foundational APIs working with spans
- Reflection emit
- SIMD (Single Instruction, Multiple Data)
- ValueTask and ValueTask<T>

Choosing the Right Version

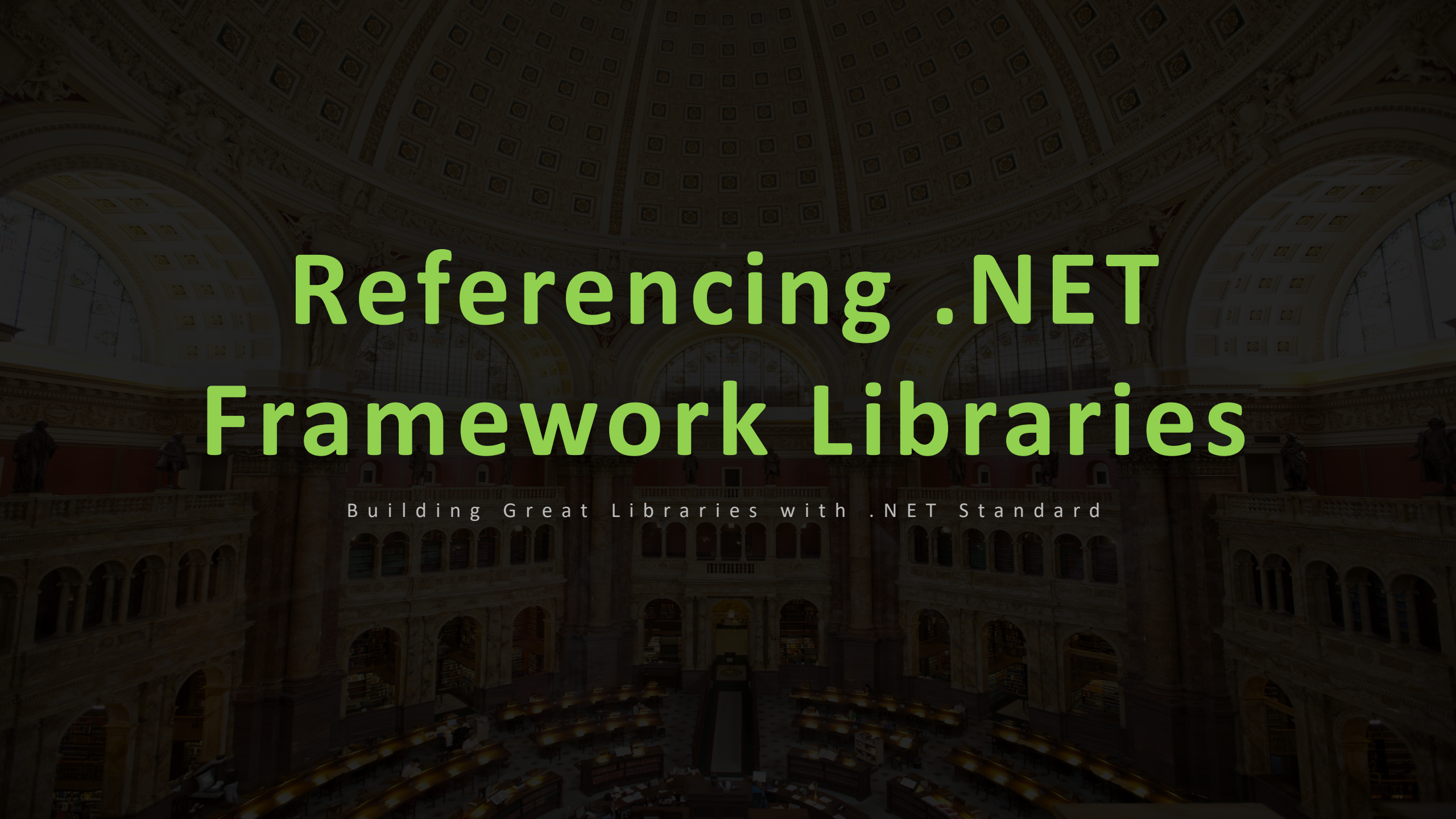
.NET Standard 2.1

- Span<T>
- Foundational APIs working with spans
- Reflection emit
- SIMD (Single Instruction, Multiple Data)
- ValueTask and ValueTask<T>
- DBProviderFactories

Choosing the Right Version

.NET Standard 2.1

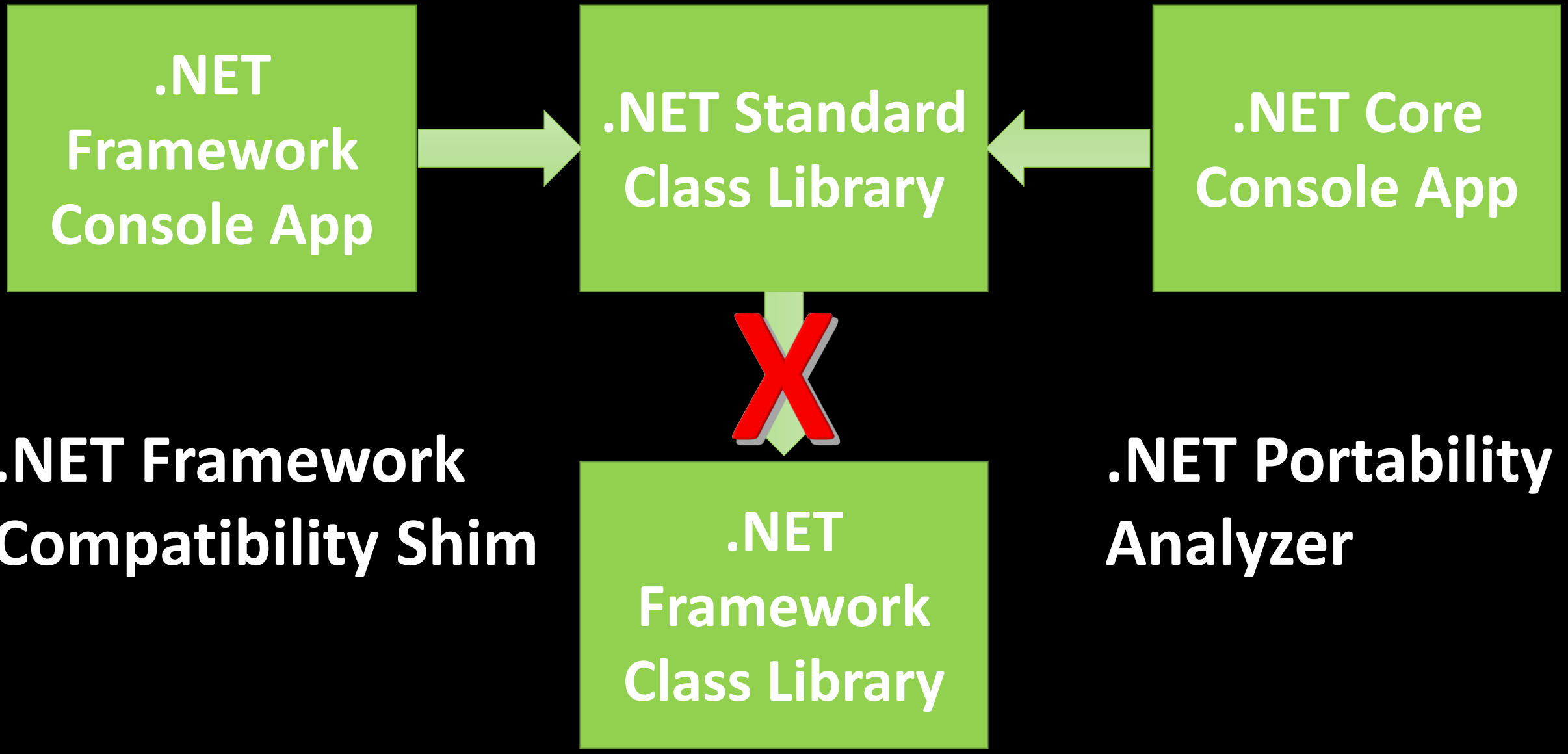
- Span<T>
- Foundational APIs working with spans
- Reflection emit
- SIMD (Single Instruction, Multiple Data)
- ValueTask and ValueTask<T>
- DBProviderFactories
- General Goodness



Referencing .NET Framework Libraries

Building Great Libraries with .NET Standard

Referencing .NET Framework Libraries



.NET Framework Compatibility Shim

.NET Portability Analyzer

When to Use .NET Standard

Building Great Libraries with .NET Standard

When to Use .NET Standard – New

Always?

When to Use .NET Standard – Migration

Target lowest possible version

Library is .NET Standard Compatible

Need to use it on different .NET Platforms

Already reference it in a .NET Standard class library

Platform Specific Code

Building Great Libraries with .NET Standard

Windows Compatibility Pack

- Microsoft.Windows.Compatibility (NuGet package)
  - Can be referenced from .NET Core as well as .NET Standard
  - Has ~21k APIs (Windows-only as well as cross-platform)
- Contents

ACLs
Code Pages
Code Dom
Configuration
Crypto
DirectoryServices

Drawing
EventLog
MEF v1
ODBC
Perf Counters
Permissions

Ports
Registry
Runtime Caching
WCF
Windows Services

Multi-Targeting Best Practices

- DO start with the .NET Standard 2.0
  - Most general purpose libraries will not need APIs outside this set
- CONSIDER targeting multiple frameworks
  - If you need to call platform-specific APIs outside of .NET Standard
- DO NOT drop support for .NET Standard
  - Instead, throw from the implementation and offer capability APIs. This way, your library can be used anywhere and supports runtime light-up.
- DO share your component using a NuGet package
  - It shields consumers from having to pick the appropriate implementation

Guidelines for Great Libraries

Building Great Libraries with .NET Standard

Versioning

Kind	When to increment	Comment
Package Version	Every change	The ID of the NuGet package.
Assembly Version	As you see fit	The version number of the assembly. Used by the loader to resolve assemblies.
File Version	Every change	Generic concept, used by installers to determine which file is newer.
Informational Version	As you see fit	Display string, does not need to be a version.

Best Practices

- DO follow the API Design Guidelines
- Do target .NET Standard 2.0
- CONSIDER using multi-targeting to allow for platform-specific code
  - CONSIDER dual-targeting for .NET Framework 4.6.1
  - DO use NuGet for packaging multi-targeted libraries
  - DO throw PlatformNotSupportedExecption for unsupported APIs
- DO strong name your libraries
  - AVOID if your library cannot be used on .NET Framework

Questions

Building Great Libraries with .NET Standard

chadgreen@chadgreen.com

chadgreen.com

ChadGreen

ChadwickEGreen

Slides will be available on my site