

.NET Framework, .NET Core & the new .NET SDK

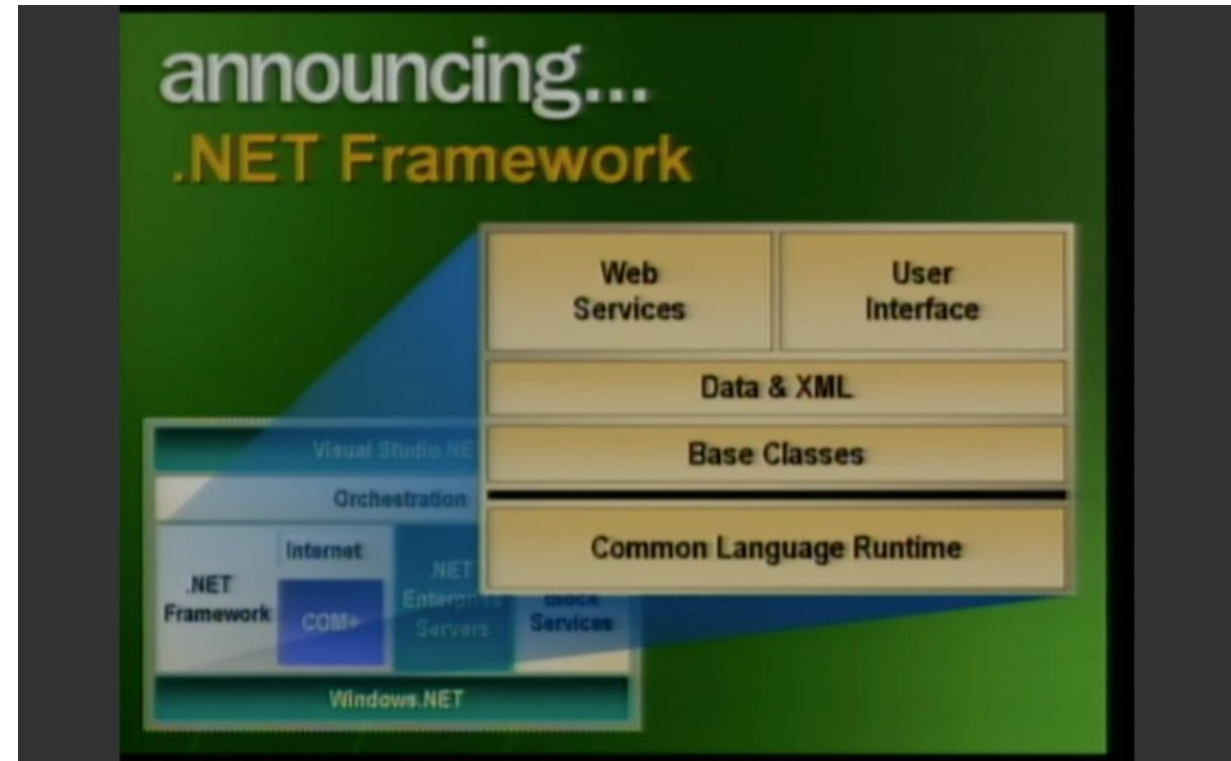
Dominick Baier
<http://leastprivilege.com>
[@leastprivilege](#)

Brock Allen
<http://brockallen.com>
[@brockallen](#)

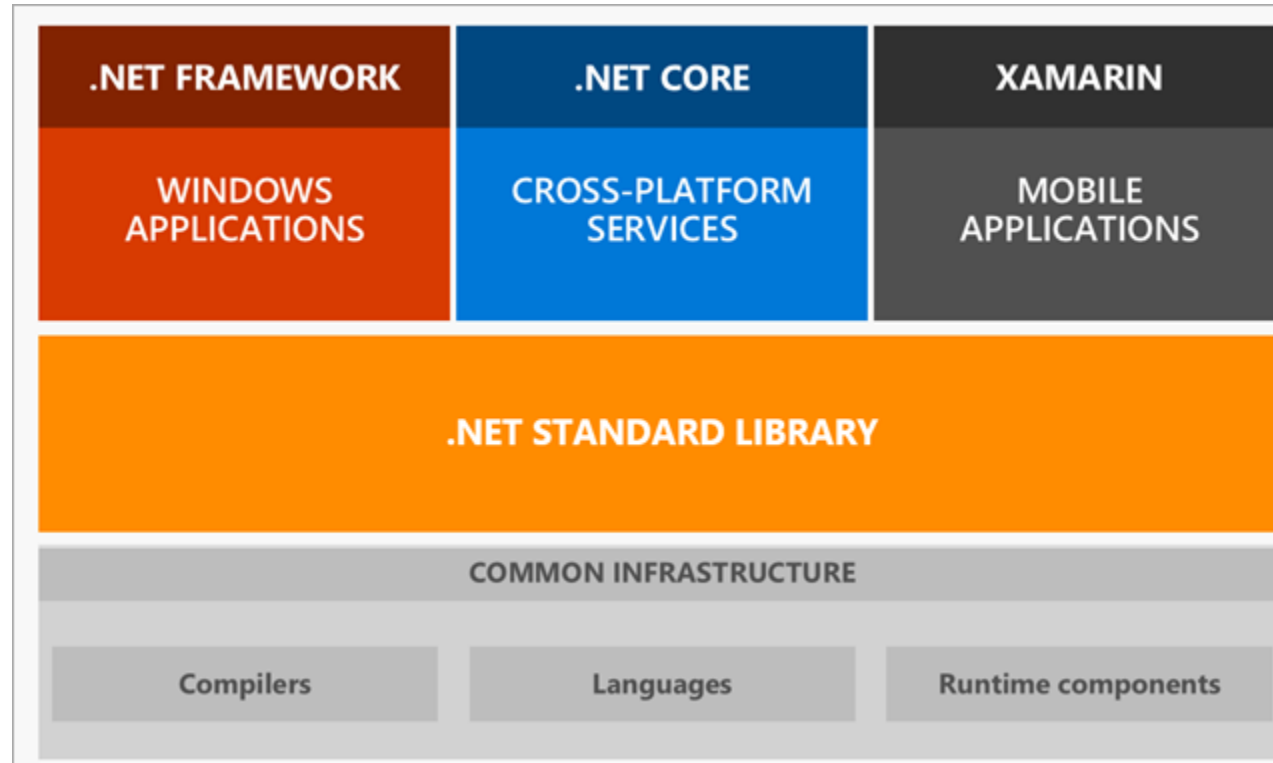


Motivation

- .NET has a strong history
 - Very popular
 - Lots of investments
- There is more than just Windows
 - Many more platforms, devices, and clouds
- .NET is evolving
 - .NET needs to learn to run in more places
 - .NET needs modern tooling



The new .NET



.NET Core : next gen .NET

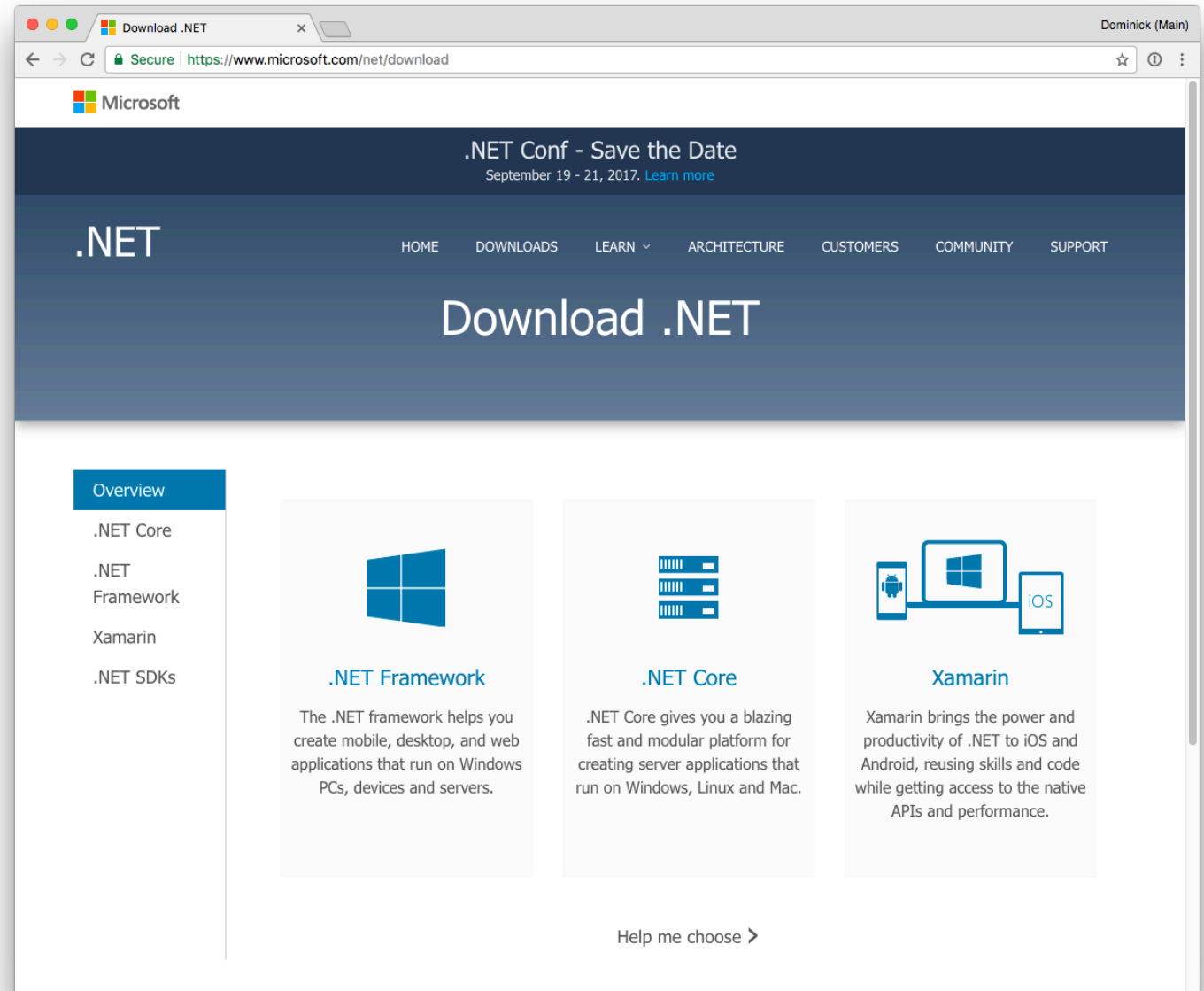
- Cross platform
 - Windows, Linux, Mac, FreeBSD
- Portable
 - Can be ~/bin deployed
 - Can be user or machine installed as well
- Open source
 - <https://github.com/dotnet/coreclr>
 - Contains core runtime and mscorlib (e.g. GC, JIT, BCL)
 - Does not contain many frameworks (e.g. WCF, WPF)

Development ecosystem

- SDK
 - Command-line tooling (*dotnet*)
- Project system
 - Simplified, file-system based project system
 - Runtime, libraries, and packaging
 - NuGet-focused
- Editors/IDEs
 - Any text editor (VS Code, Emacs, Sublime, etc) and OmniSharp (OSS)
 - Visual Studio (Microsoft)
 - Project Rider (JetBrains)

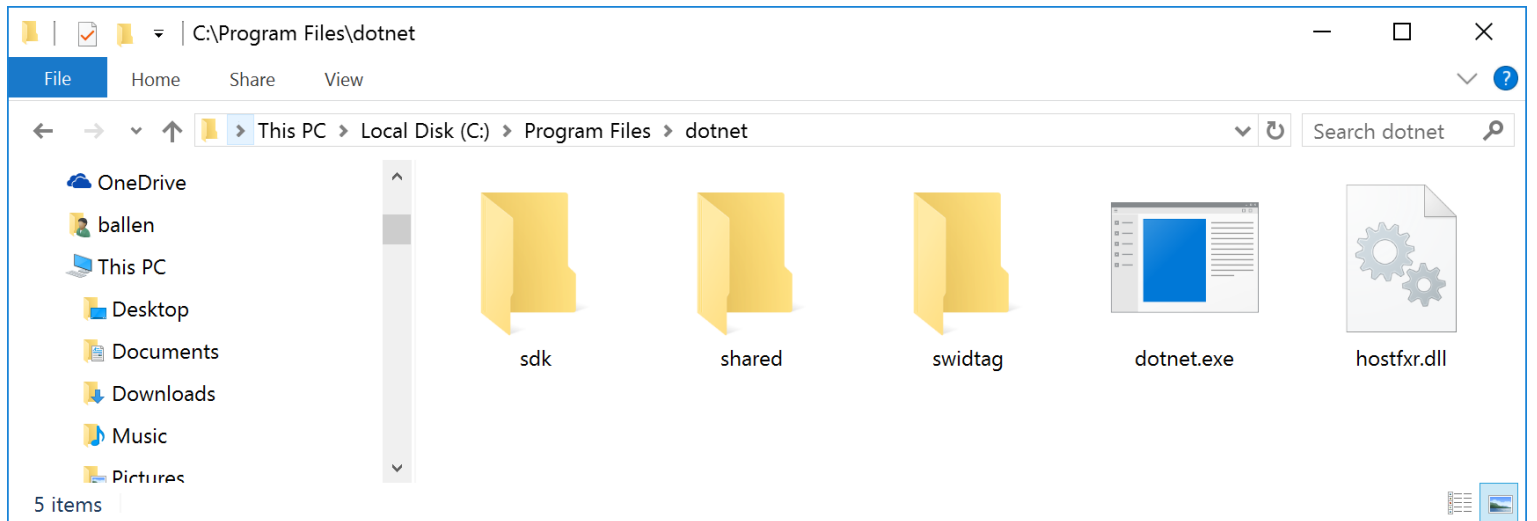
Installing .NET SDK

- Visit <https://dot.net>
 - Installer
 - Docs
 - Tutorials
 - etc...



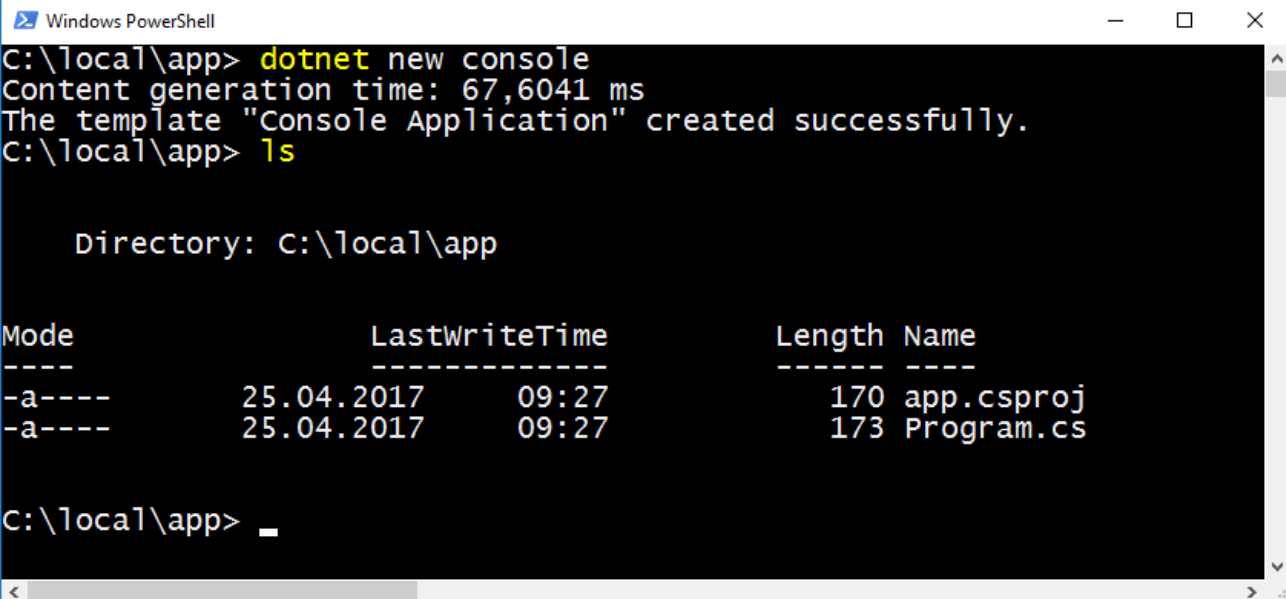
Command line interface: *dotnet*

- Create new project
- Install NuGet dependencies
- Build application
- Load .NET and run application
- Testing
- Package library
- Publish application



dotnet new

- Creates new project
 - program.cs
 - app.csproj
- Console-based application



```
Windows PowerShell
C:\local\app> dotnet new console
Content generation time: 67,6041 ms
The template "Console Application" created successfully.
C:\local\app> ls

Directory: C:\local\app

Mode                LastWriteTime         Length Name
----                -
-a----             25.04.2017     09:27         170 app.csproj
-a----             25.04.2017     09:27         173 Program.cs

C:\local\app> _
```

```
using System;

namespace app
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello world!");
        }
    }
}
```


.csproj

- Simplified MsBuild file
 - Compiler settings
 - Target framework
 - File globbing

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>

    <OutputType>Exe</OutputType>
    <TargetFramework>netcoreapp2.0</TargetFramework>

  </PropertyGroup>
</Project>
```

Dependencies

- dotnet add
 - nuget packages
 - project references

```
dotnet add package newtonsoft.json
```

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>

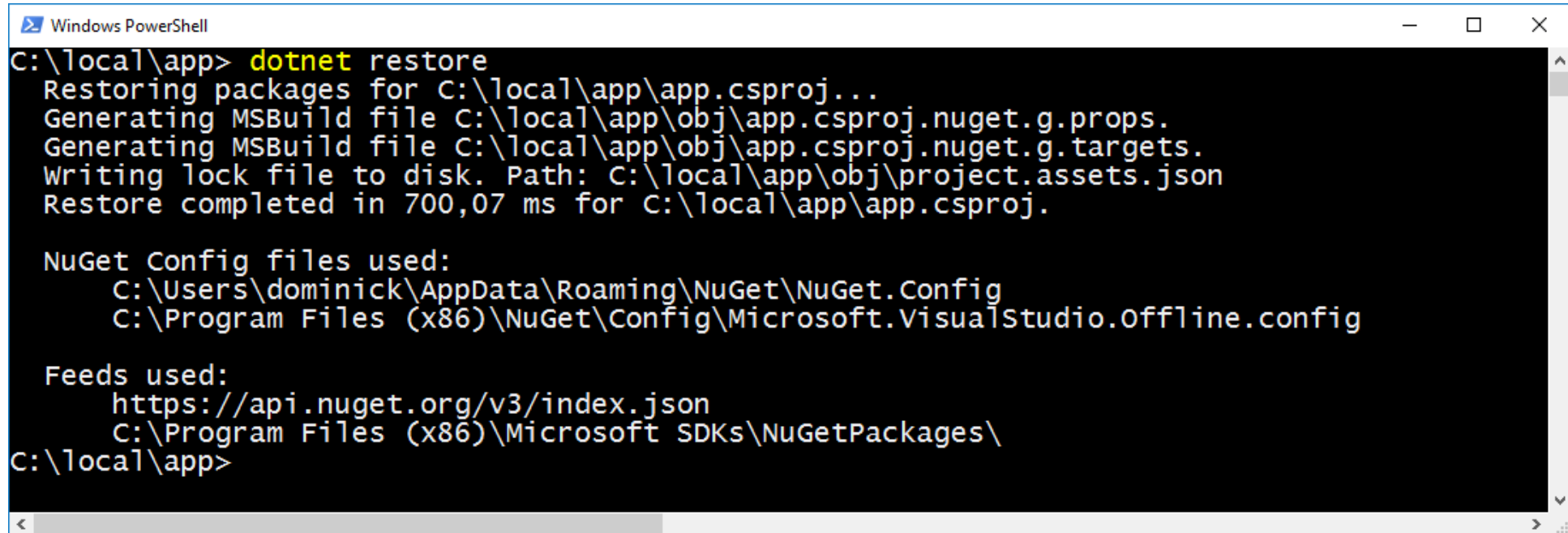
    <OutputType>Exe</OutputType>
    <TargetFramework>netcoreapp2.0</TargetFramework>

  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="newtonsoft.json"
                      Version="10.0.2" />
  </ItemGroup>
</Project>
```

dotnet restore

- Downloads NuGet packages
- Builds lock file
 - Snapshot of dependency versions



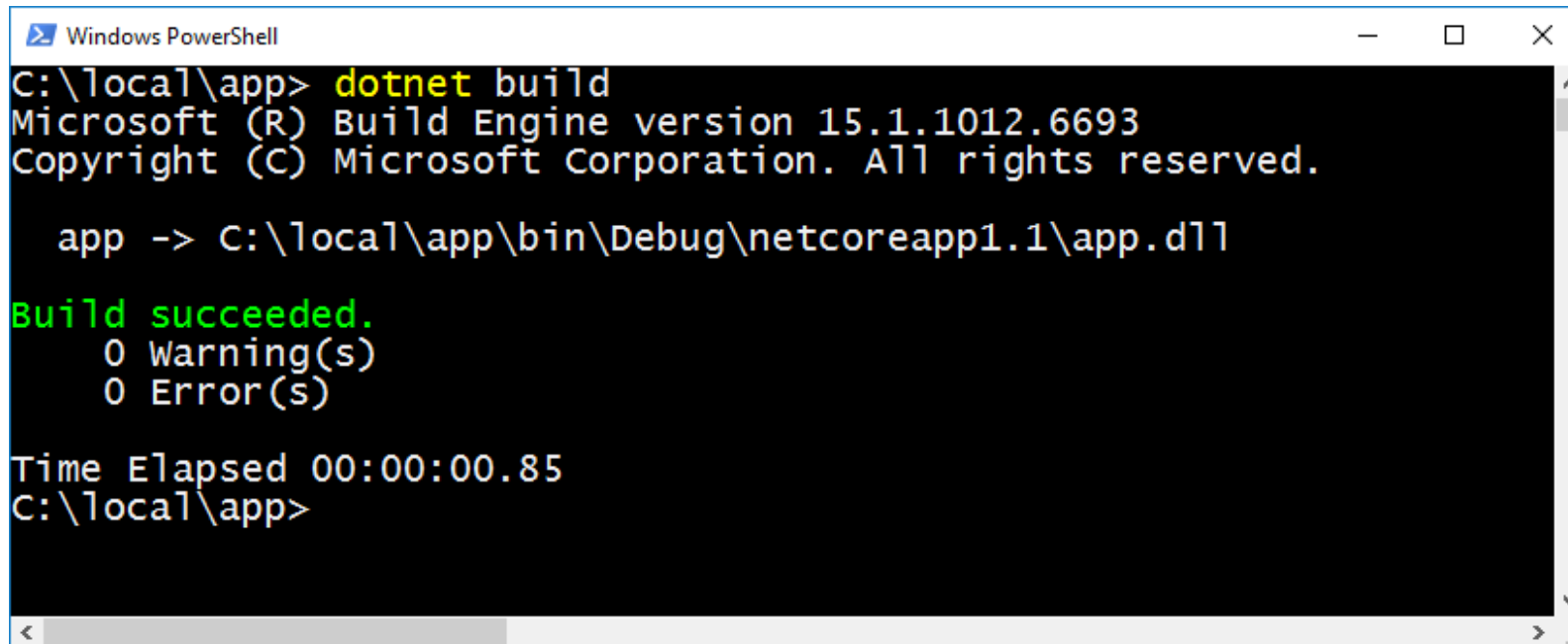
```
Windows PowerShell
C:\local\app> dotnet restore
Restoring packages for C:\local\app\app.csproj...
Generating MSBuild file C:\local\app\obj\app.csproj.nuget.g.props.
Generating MSBuild file C:\local\app\obj\app.csproj.nuget.g.targets.
Writing lock file to disk. Path: C:\local\app\obj\project.assets.json
Restore completed in 700,07 ms for C:\local\app\app.csproj.

NuGet Config files used:
  C:\Users\dominick\AppData\Roaming\NuGet\NuGet.Config
  C:\Program Files (x86)\NuGet\Config\Microsoft.VisualStudio.Offline.config

Feeds used:
  https://api.nuget.org/v3/index.json
  C:\Program Files (x86)\Microsoft SDKs\NuGetPackages\
C:\local\app>
```

dotnet build / dotnet run / dotnet app.dll

- Builds project, or builds and runs application
- Binaries output to *~/bin/<configuration>/<framework>* folder



```
Windows PowerShell
C:\local\app> dotnet build
Microsoft (R) Build Engine version 15.1.1012.6693
Copyright (C) Microsoft Corporation. All rights reserved.

    app -> C:\local\app\bin\Debug\netcoreapp1.1\app.dll

Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:00.85
C:\local\app>
```

Library projects: *dotnet new classlib*

- Creates library project
- Targets ".NET platform standard"
 - "netstandardX.Y" TFM
 - Allows libraries to target multiple .NET frameworks (current and future)

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>

    <TargetFramework>netstandard2.0</TargetFramework>

  </PropertyGroup>
</Project>
```

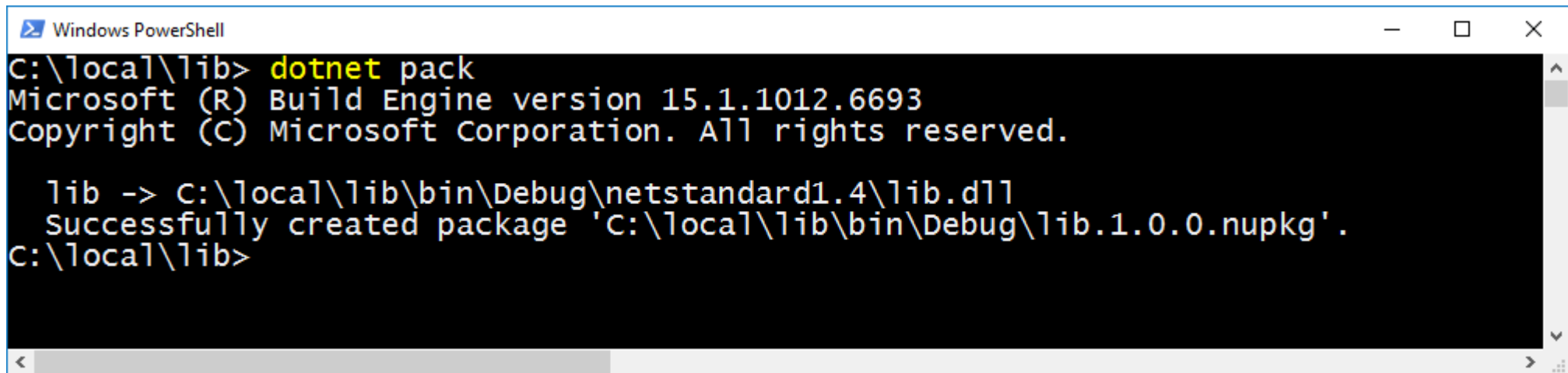
.NET implementation support

The following table lists all versions of .NET Standard and the platforms supported:

| .NET Standard | 1.0 | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 2.0 |
|---|------|------|-------|------|-------|-------|-------|-------|
| .NET Core | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 2.0 |
| .NET Framework (with .NET Core 1.x SDK) | 4.5 | 4.5 | 4.5.1 | 4.6 | 4.6.1 | 4.6.2 | | |
| .NET Framework (with .NET Core 2.0 SDK) | 4.5 | 4.5 | 4.5.1 | 4.6 | 4.6.1 | 4.6.1 | 4.6.1 | 4.6.1 |
| Mono | 4.6 | 4.6 | 4.6 | 4.6 | 4.6 | 4.6 | 4.6 | 5.4 |
| Xamarin.iOS | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.14 |
| Xamarin.Mac | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.8 |
| Xamarin.Android | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.5 |
| Universal Windows Platform | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | vNext | vNext | vNext |
| Windows | 8.0 | 8.0 | 8.1 | | | | | |
| Windows Phone | 8.1 | 8.1 | 8.1 | | | | | |
| Windows Phone Silverlight | 8.0 | | | | | | | |

dotnet pack

- Packages library into NuGet



```
Windows PowerShell
C:\local\lib> dotnet pack
Microsoft (R) Build Engine version 15.1.1012.6693
Copyright (C) Microsoft Corporation. All rights reserved.

lib -> C:\local\lib\bin\Debug\netstandard1.4\lib.dll
Successfully created package 'C:\local\lib\bin\Debug\lib.1.0.0.nupkg'.
C:\local\lib>
```

Publishing

- Preparing application for deployment
 - Including dependencies (i.e. libraries and possibly the .NET framework)
- Portable
 - Application assumes .NET framework already installed on target platform
 - Application can run on any platform with the target framework
- Self-contained
 - Application ships .NET framework with application
 - Must build for a specific platform

dotnet publish

- Prepares application for deployment

The image shows a Windows PowerShell window and a File Explorer window. The PowerShell window displays the command `dotnet publish` and its output, which lists the paths for the published application and its dependencies. The File Explorer window shows the resulting files in the `publish` directory, including `app.deps.json`, `app.dll`, `app.pdb`, `app.runtimeconfig.json`, `lib.dll`, `lib.pdb`, and various system DLLs.

Windows PowerShell Output:

```
C:\local\app> dotnet publish
Microsoft (R) Build Engine version 15.1.1012.6693
Copyright (C) Microsoft Corporation. All rights reserved.

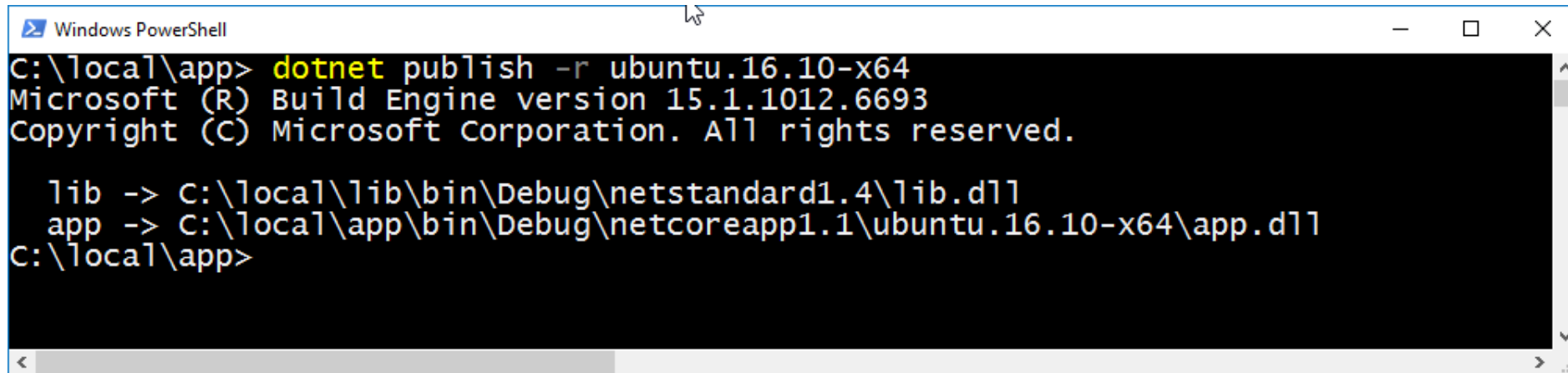
lib -> C:\local\lib\bin\Debug\netstandard1.4\lib.dll
app -> C:\local\app\bin\Debug\netcoreapp1.1\app.dll
C:\local\app>
```

File Explorer Contents:

| Name | Date modified | Type | Size |
|---|------------------|-----------------------|--------|
| app.deps.json | 25.04.2017 12:05 | JSON File | 6 KB |
| app.dll | 25.04.2017 12:05 | Application extens... | 5 KB |
| app.pdb | 25.04.2017 12:05 | Program Debug D... | 1 KB |
| app.runtimeconfig.json | 25.04.2017 12:05 | JSON File | 1 KB |
| lib.dll | 25.04.2017 12:02 | Application extens... | 4 KB |
| lib.pdb | 25.04.2017 12:02 | Program Debug D... | 1 KB |
| Newtonsoft.Json.dll | 02.04.2017 17:46 | Application extens... | 622 KB |
| System.Collections.NonGeneric.dll | 07.03.2017 19:21 | Application extens... | 87 KB |
| System.Collections.Specialized.dll | 07.03.2017 19:21 | Application extens... | 45 KB |
| System.ComponentModel.Primitives.dll | 07.03.2017 19:21 | Application extens... | 36 KB |
| System.ComponentModel.TypeConverte... | 07.03.2017 19:21 | Application extens... | 119 KB |
| System.Runtime.Serialization.Formatter... | 05.11.2016 04:56 | Application extens... | 31 KB |
| System.Runtime.Serialization.Primitives.dll | 07.03.2017 19:21 | Application extens... | 29 KB |
| System.Xml.XmlDocument.dll | 05.11.2016 04:57 | Application extens... | 135 KB |

Self-contained publishing

- Add runtime identifier to publish command
 - <https://github.com/dotnet/docs/blob/master/docs/core/rid-catalog.md>



```
Windows PowerShell
C:\local\app> dotnet publish -r ubuntu.16.10-x64
Microsoft (R) Build Engine version 15.1.1012.6693
Copyright (C) Microsoft Corporation. All rights reserved.

lib -> C:\local\lib\bin\Debug\netstandard1.4\lib.dll
app -> C:\local\app\bin\Debug\netcoreapp1.1\ubuntu.16.10-x64\app.dll
C:\local\app>
```

Summary

- Command-line first
 - makes automating builds much easier
 - same cli for all platforms
- Simplified project format
- Nuget as a first-class citizen