

Get Comfortable with .NET 5 and the CLI

Jeremy Clark

www.jeremybytes.com

@jeremybytes

What is .NET 5?

- Formerly .NET Core
- The new face of .NET
- Quickly evolving & open source
- Cross-platform (Windows, macOS, Linux)

What I like about .NET 5

- .NET 5 is easy to get installed and configured
- Self-hosted web applications with Kestrel
- Vastly simplified ASP.NET templates
- Built-in Dependency Injection works for many scenarios

What I like about .NET 5

- Command-line interface (CLI)
- Command-line help is actually helpful
- The same tool chain can be used across platforms (CLI, Visual Studio Code)
- Cross-platform actually works! (great for containers)

Why .NET 5?

- .NET 5 is the future
- C# 8+ language features will *NOT* come to .NET Framework
- .NET Framework is “done” (4.8 is the last version)

What you need to build applications with .NET 5

- .NET 5 SDK

<https://dotnet.microsoft.com/download>

- Visual Studio Code (or Visual Studio)

<https://code.visualstudio.com/download>

- C# Extension for Visual Studio Code

<https://marketplace.visualstudio.com/items?itemName=ms-vscode.csharp>

CLI: Command-line Interface

- `dotnet`
- `dotnet new`
- `dotnet add package`
- `dotnet build`
- `dotnet run`

dotnet -h

```
PS D:\> dotnet -h
.NET SDK (5.0.301)
Usage: dotnet [runtime-options] [path-to-application] [arguments]
```

Execute a .NET application.

runtime-options:

<code>--additionalprobingpath <path></code>	Path containing probing policy and assemblies to probe
<code>--additional-deps <path></code>	Path to additional deps.json file.
<code>--depsfile</code>	Path to <application>.deps.json file.
<code>--fx-version <version></code>	Version of the installed Shared Framework to use to run
<code>--roll-forward <setting></code>	Roll forward to framework version (LatestPatch, Minor, NoRollForward)
<code>--runtimeconfig</code>	Path to <application>.runtimeconfig.json file.

path-to-application:

The path to an application .dll file to execute.

```
Usage: dotnet [sdk-options] [command] [command-options] [arguments]
```


dotnet -h

SDK commands:

add	Add a package or reference to a .NET project.
build	Build a .NET project.
build-server	Interact with servers started by a build.
clean	Clean build outputs of a .NET project.
help	Show command line help.
list	List project references of a .NET project.
msbuild	Run Microsoft Build Engine (MSBuild) commands.
new	Create a new .NET project or file.
nuget	Provides additional NuGet commands.
pack	Create a NuGet package.
publish	Publish a .NET project for deployment.
remove	Remove a package or reference from a .NET project.
restore	Restore dependencies specified in a .NET project.
run	Build and run a .NET project output.
sln	Modify Visual Studio solution files.
store	Store the specified assemblies in the runtime package store.
test	Run unit tests using the test runner specified in a .NET project.
tool	Install or manage tools that extend the .NET experience.
vstest	Run Microsoft Test Engine (VSTest) commands.

dotnet new

Template Name	Short Name	Language	Tags
Console Application	console	[C#], F#, VB	Common/Console
Class library	classlib	[C#], F#, VB	Common/Library
WPF Application	wpf	[C#], VB	Common/WPF
WPF Class library	wpflib	[C#], VB	Common/WPF
WPF Custom Control Library	wpfcustomcontrollib	[C#], VB	Common/WPF
WPF User Control Library	wpfusercontrollib	[C#], VB	Common/WPF
Windows Forms App	winforms	[C#], VB	Common/WinForms
Windows Forms Control Library	winformscontrollib	[C#], VB	Common/WinForms
Windows Forms Class Library	winformslib	[C#], VB	Common/WinForms
Worker Service	worker	[C#], F#	Common/Worker/Web
MSTest Test Project	mstest	[C#], F#, VB	Test/MSTest
NUnit 3 Test Project	nunit	[C#], F#, VB	Test/NUnit
NUnit 3 Test Item	nunit-test	[C#], F#, VB	Test/NUnit
xUnit Test Project	xunit	[C#], F#, VB	Test/xUnit
Razor Component	razorcomponent	[C#]	Web/ASP.NET
Razor Page	page	[C#]	Web/ASP.NET
MVC ViewImports	viewimports	[C#]	Web/ASP.NET
MVC ViewStart	viewstart	[C#]	Web/ASP.NET
Blazor Server App	blazorserver	[C#]	Web/Blazor
Blazor WebAssembly App	blazorwasm	[C#]	Web/Blazor/WebAssembly
ASP.NET Core Empty	web	[C#], F#	Web/Empty

Creating & Running

- New Projects

- `dotnet new webapi`

- `dotnet new console`

- Build / Run

- `dotnet build`

- `dotnet run`

Adding Packages / References

- NuGet Packages

```
dotnet add package Newtonsoft.Json
```

- Add Reference

```
dotnet add reference ../folder/my-library.csproj
```

Solutions

- Solution Files

```
dotnet new sln
```

```
dotnet sln add ./folder1/my-console.csproj
```

```
dotnet sln add ./folder2/my-library.csproj
```

ASP.NET Dependency Injection

- Auto-inject dependencies as parameters in Controller constructors
- In Startup.cs

```
services.AddSingleton<IPeopleProvider, CSVPeopleProvider>()
```
- Lifetimes: Singleton, Scoped, Transient



Thank You!

Jeremy Clark

- <http://www.jeremybytes.com>
- jeremy@jeremybytes.com
- @jeremybytes