Using .NET Core 3 and Desktop

Install .NET Core 3

* <https://dotnet.microsoft.com/download>
* VS 2017 will only support up to .NET Core 2.2, Visual Studio 2019 Preview is required for .NET Core 3 support
* .NET Core is side by side, to see what SDK you are running use dotnet –info
* If experience an incompatibility after installing a preview SDK project, pin your project to an particular SDK
  + dotnet --list-sdks
  + dotnet new globaljson --sdk-version [version from list above]

Migrating any referenced projects to SDK style projects

* See this guide: https://natemcmaster.com/blog/2017/03/09/vs2015-to-vs2017-upgrade/
* Open the .csproj file and replace it with:

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

<PropertyGroup>

<TargetFramework>net472</TargetFramework>

<GenerateAssemblyInfo>false</GenerateAssemblyInfo>

</PropertyGroup>

</Project>

Compile the project to see if it works. Once it works move it to .NET Standard 2.0

* Open the .csproj file set the target framework to .NET Standard 2.0

<TargetFramework>netstandard2.0</TargetFramework>

* Compile the application, if there are missing API’s you can look and see if the Windows Compatibility Pack can help: <https://blogs.msdn.microsoft.com/dotnet/2017/11/16/announcing-the-windows-compatibility-pack-for-net-core/>
* Add the Nuget package: Microsoft.Windows.Compatibility

Check API Compatibility

* Install the package Microsoft.DotNet.Analyzers.Compatibility
* Error List | Warnings will now show API’s that are not supported on other platforms

Create .NET Core 3 Windows Form App

* dotnet new winforms -o [name of a app]
* Include the source files from the other project by opening .csproj

<ItemGroup>

<Compile Include="..\[folder of .net fx app]\\*\*\\*.cs" />

</ItemGroup>

* Turn off AssemblyInfo generation: <GenerateAssemblyInfo>false</GenerateAssemblyInfo>
* Reference any libraries

Blazor

* Open Solution in VS 2017
* Add the two Blazor projects to the solution
* Set the Blazor server as the startup project
* Now you can see the same app, using the same .NET Standard 2 library in the browser