Microsoft .NET Framework

Windows Programming Course

Agenda

1. What is .NET Framework?
2. Architecture of .NET Framework
3. .NET Framework and .NET Core
4. Using Visual Studio

1.

What is .NET Framework?

What is .NET Framework?

- The .NET Framework is:
  - A run-time execution environment that manages apps that target .NET Framework.
  - Provide tools and libraries that enable developer to create windows applications, web services or web applications in variety of programming languages such as C++, C#, F#, VB…

Who uses .NET Framework?

Users of applications

Users of applications built with the .NET Framework need to have **.NET Framework Runtime** installed.

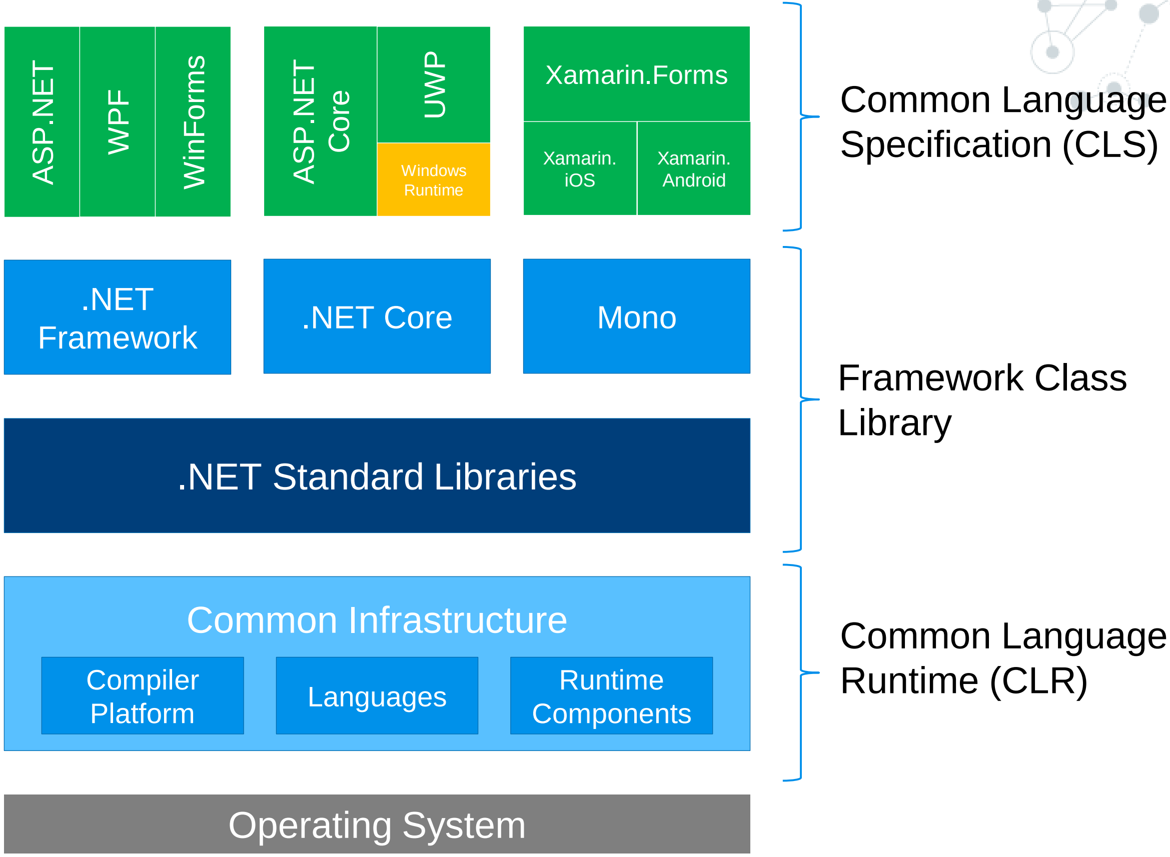
Developers

Developers use **the developer pack** (SDK – Software develop kit) to create applications that run on .NET Framework.

2.

Characteristics and architecture

Architecture of .NET Framework

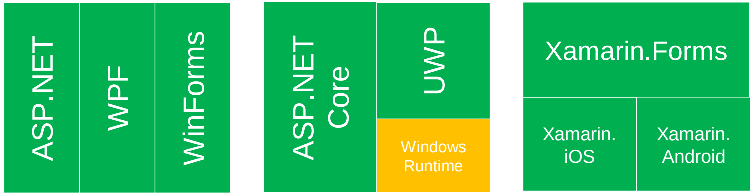


Architecture – Operating System

- Operating System: typically is Microsoft Windows, provide Win32 API (Application Programming Interface)

Architecture – Common Language Specification (CLS)

- The CLS is a common platform that integrates code and components from multiple .NET programming languages.



ASP.NET

WPF

WinForms

ASP.NET Core

UWP

Windows Runtime

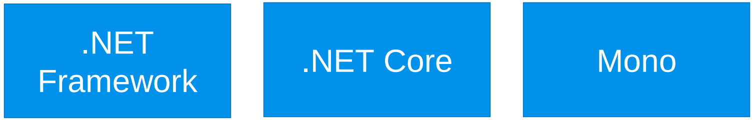
Xamarin.Forms

Xamarin. iOS

Xamarin. Android

Architecture – Framework Class Library (FCL)

- The FCL provides the system functionality in the .NET Framework as it has various classes, data types, interfaces, etc. to perform multiple functions and build different types of applications such as desktop applications, web applications, mobile applications, etc.

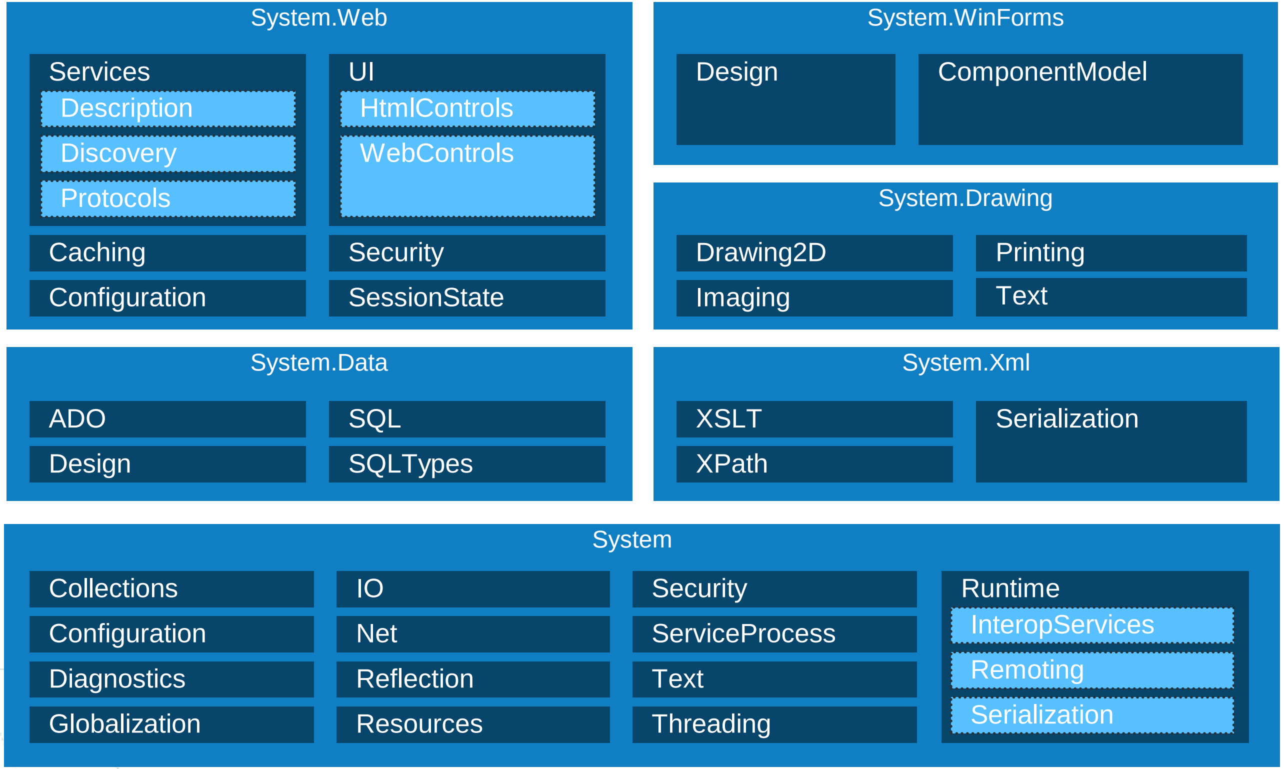


.NET Framework

.NET Core

Mono

Architecture – Framework Class Library (FCL)



System.Web
- Services
  - Description
  - Discovery
  - Protocols
- UI
  - HtmlControls
  - WebControls
- Caching
- Security
- Configuration
- SessionState

System.WinForms
- Design
- ComponentModel

System.Drawing
- Drawing2D
- Printing
- Imaging
- Text

System.Data
- ADO
- SQL
- Design
- SQLTypes

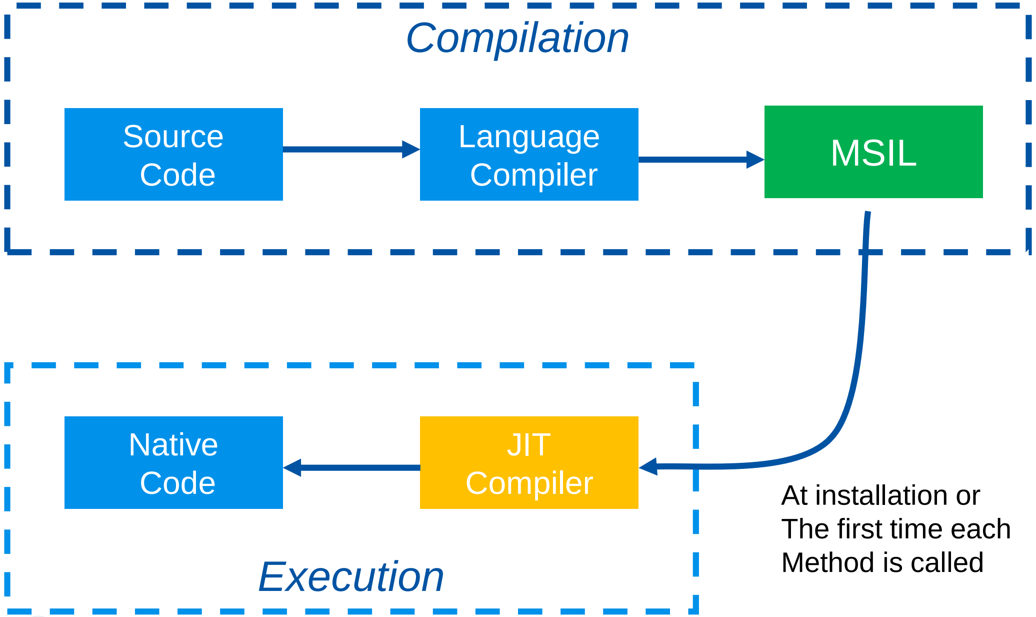
System.Xml
- XSLT
- Serialization
- XPath

System
- Collections
- IO
- Security
- Runtime
  - InteropServices
  - Remoting
  - Serialization
- Configuration
- Net
- ServiceProcess
- Diagnostics
- Reflection
- Text
- Globalization
- Resources
- Threading

Architecture – Common Language Runtime (CLR)

- The CLR is the excution engine for .NET application and serves as the interface between .NET Applications and the OS.
- The CLR provides many services such as:
  - Compiles and executes code
  - Convert intermediate language to native machine code
  - Manage memory, processes
  - …

CLR – Compilation and Execution



At installation or
The first time each
Method is called

Microsoft Intermediate Language (MSIL)

CPU independent set of instruction.

Can be efficiently converted to native code.

Just-in-time compiler (JIT)

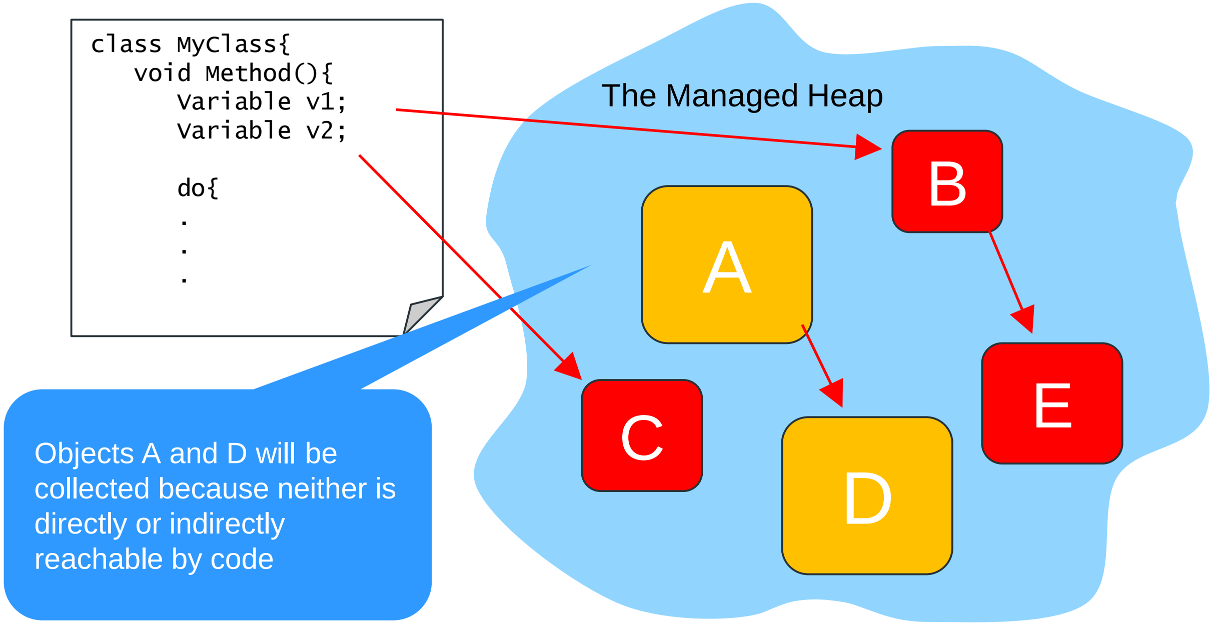
Compiled at runtime

Convert MSIL to native code

CLR – Automatic Memory Management

- The CLR manages memory for managed code
  - All allocations of objects and buffers made from a Managed Heap
  - Unused objects and buffers are cleaned up automatically through Garbage Collection
- Some of the worst bugs in software development are not possible with managed code
  - Buffer overflows (50% of security bugs)
  - Leaked memory or objects
  - References to freed or non-existent objects
  - Reading of uninitialized variables

CLR – Garbage Collection



```
class MyClass{
   void Method(){
      Variable v1;
      Variable v2;

      do{
      .
      .
      .
```

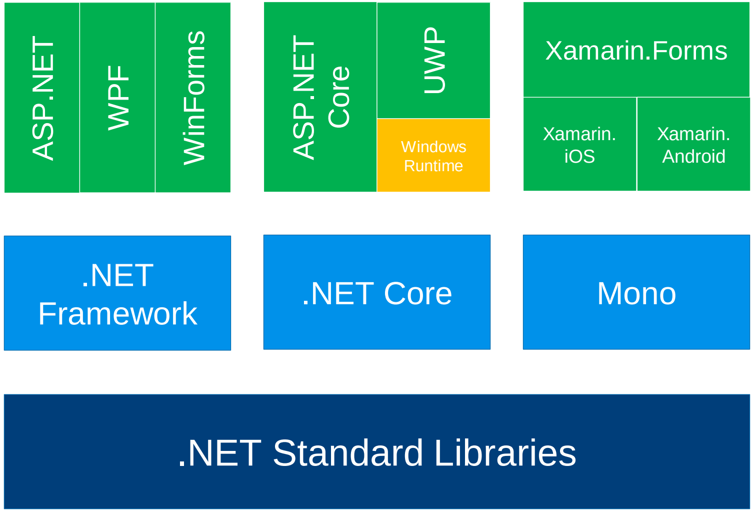
Objects A and D will be collected because neither is directly or indirectly reachable by code

🟥 = Object in memory

3.

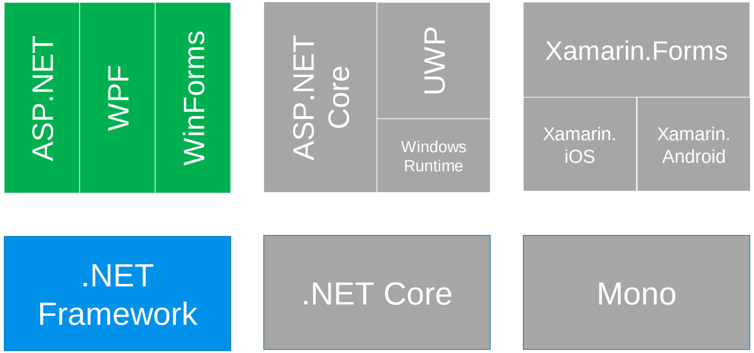
.NET Framework and .NET Core

.NET Framework and .NET Core



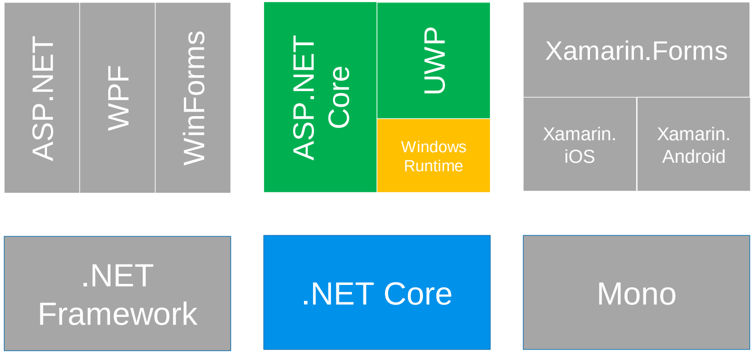
.NET Framework

- Used to create Windows Forms, WPF, and legacy ASP.NET applications that *run on Windows*
- Continuously enhanced in the past 15 years



.NET Core

- Open-source – find it at https://github.com/dotnet
- Designed in a modular approach: the framework splits up into a large list of NuGet packages



ASP.NET

WPF

WinForms

.NET Framework

ASP.NET Core

UWP

Windows Runtime

.NET Core

Xamarin.Forms

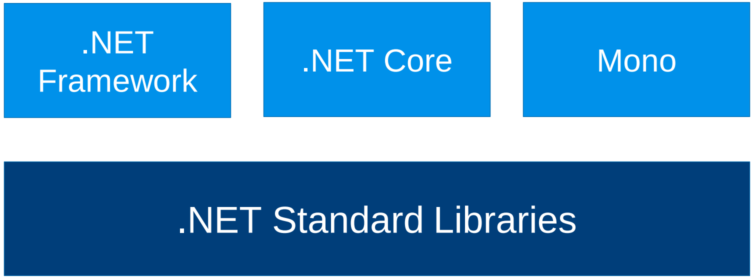
Xamarin. iOS

Xamarin. Android

Mono

.NET Standard

- It’s a contract (not an implementation)
- Specify what APIs need to be implemented .NET Framework, .NET Core.



.NET Framework

.NET Core

Mono

.NET Standard Libraries

4.

Using Visual Studio

Integrated Development Environment (IDE)

IDE for C# programming

- Visual Studio
- Visual Code

Choosing Visual Studio editions

Community

Free for open-source projects, training, academic and small professional teams.

Professional

Has more features, such as CodeLens and Team Foundation Server for source code management and team collaboration.

Enterprise

Contains a lot of tools for testing, such as Web Load & Performance Testing, Unit Test Isolation with Microsoft Fakes, and Coded UI Testing.

Installing Visual Studio

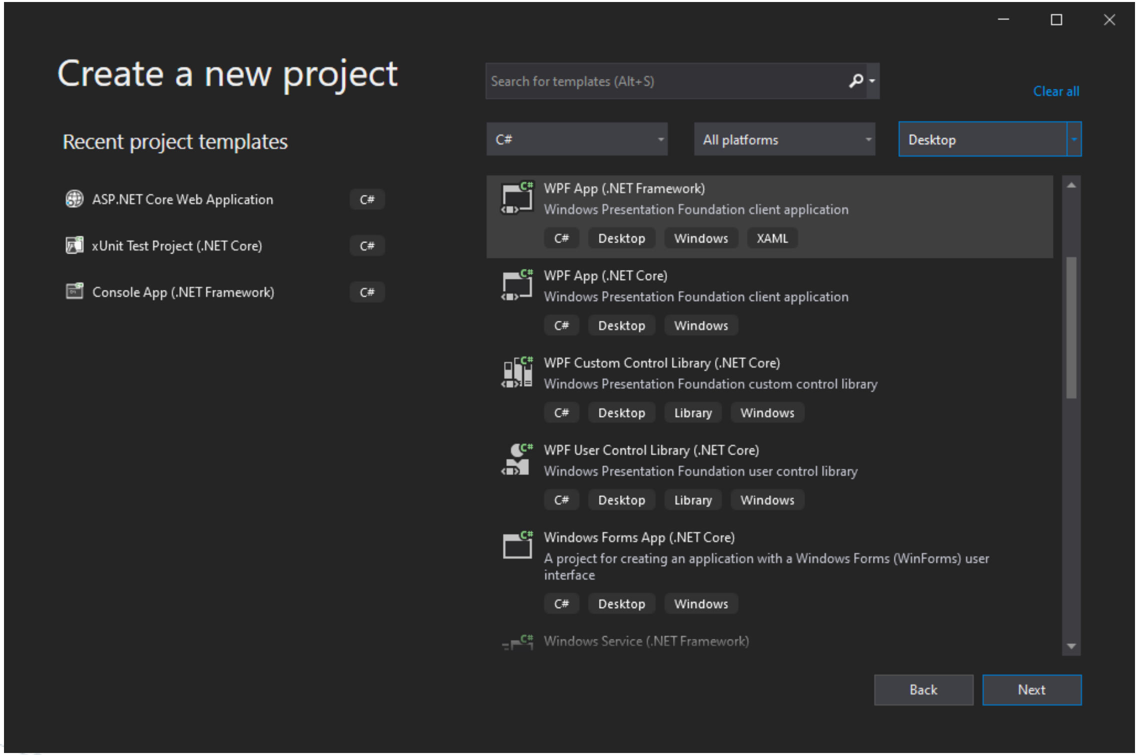
- Be able to install necessary workloads:
  - Universal Windows Platform development .NET Desktop development
  - ASP.NET and web development
  - Azure development
  - Mobile development with .NET
  - .NET Core cross-platform development

Hands-on Exercise

Download and install following software:

- Visual Studio 2019 (Community)
- SQL Server 2019 Express

Creating a Project



Create a new project

Recent project templates

ASP.NET Core Web Application C#

xUnit Test Project (.NET Core) C#

Console App (.NET Framework) C#

Search for templates (Alt+S)

Clear all

C# | All platforms | Desktop

WPF App (.NET Framework)
Windows Presentation Foundation client application
C# Desktop Windows XAML

WPF App (.NET Core)
Windows Presentation Foundation client application
C# Desktop Windows

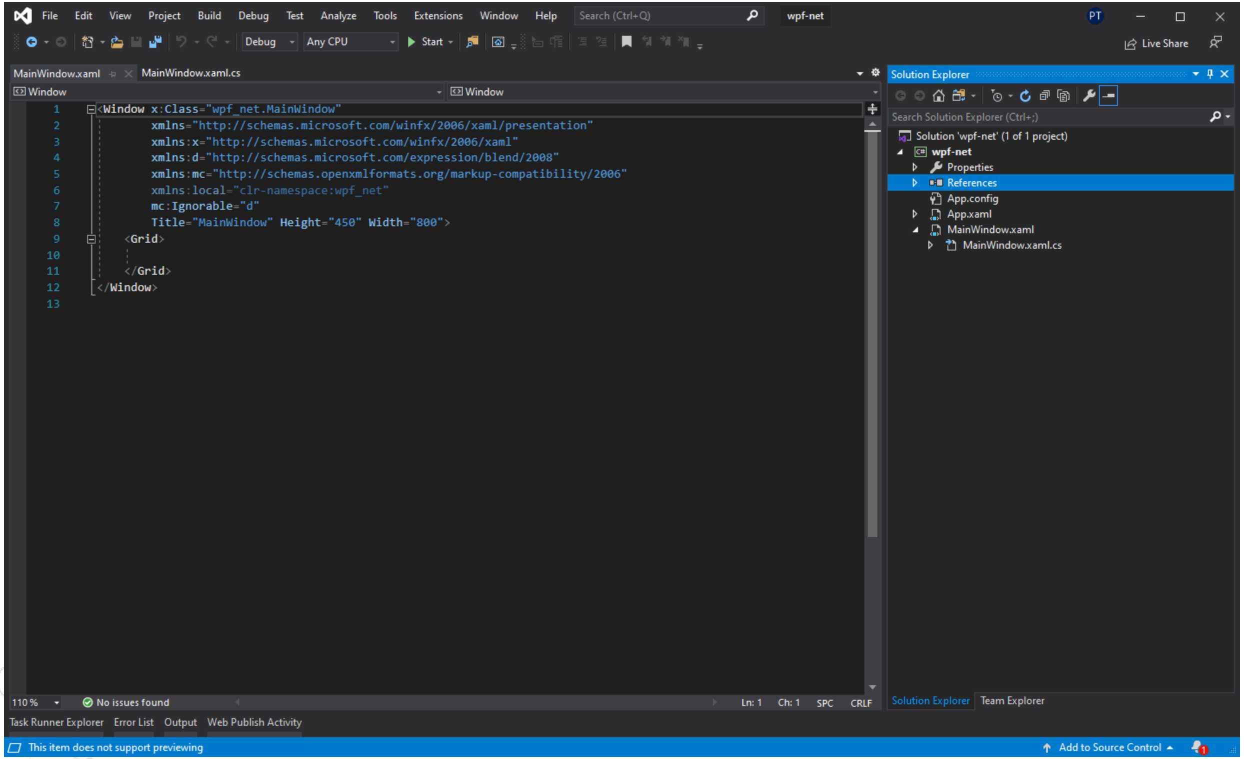
WPF Custom Control Library (.NET Core)
Windows Presentation Foundation custom control library
C# Desktop Library Windows

WPF User Control Library (.NET Core)
Windows Presentation Foundation user control library
C# Desktop Library Windows

Windows Forms App (.NET Core)
A project for creating an application with a Windows Forms (WinForms) user interface
C# Desktop Windows

Windows Service (.NET Framework)

Working with a Project



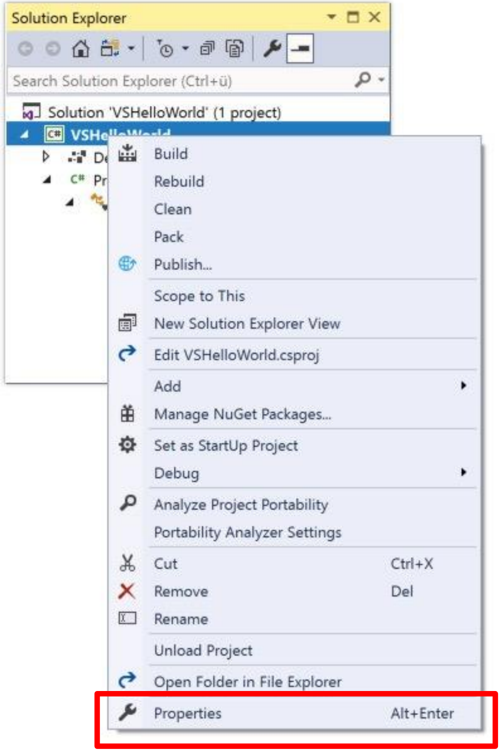
```
<Window x:Class="wpf_net.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:wpf_net"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800">
    <Grid>

    </Grid>
</Window>
```

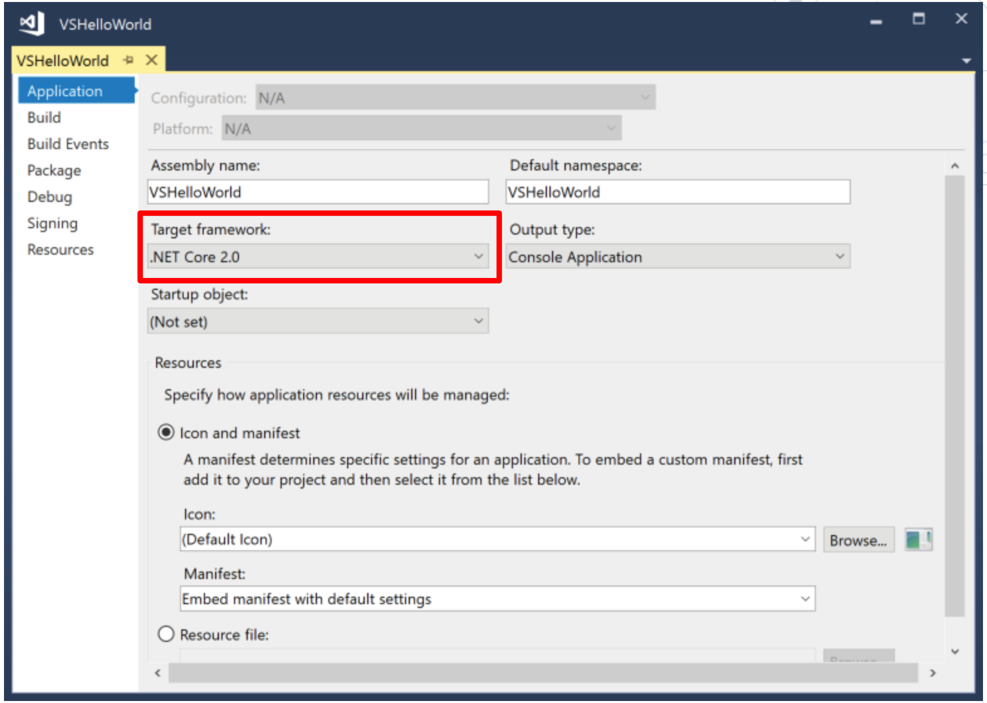
Solution 'wpf-net' (1 of 1 project)
- wpf-net
  - Properties
  - References
  - App.config
  - App.xaml
  - MainWindow.xaml
    - MainWindow.xaml.cs

No issues found

This item does not support previewing

Configuring Project Properties





Configuration: N/A

Platform: N/A

Assembly name: VSHelloWorld

Default namespace: VSHelloWorld

Target framework: .NET Core 2.0

Output type: Console Application

Startup object: (Not set)

Resources

Specify how application resources will be managed:

● Icon and manifest

A manifest determines specific settings for an application. To embed a custom manifest, first add it to your project and then select it from the list below.

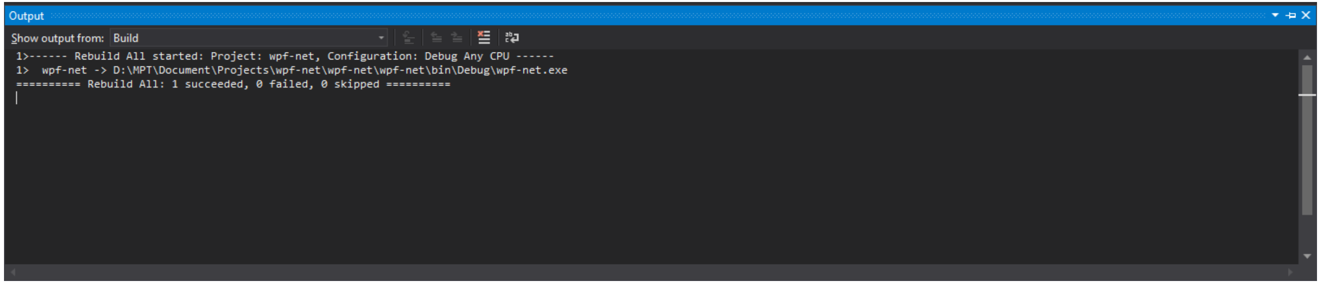
Icon: (Default Icon)

Manifest: Embed manifest with default settings

○ Resource file:

Building and running a Project

- Build ➪ Build Solution
- Debug ➪ Start Without Debugging

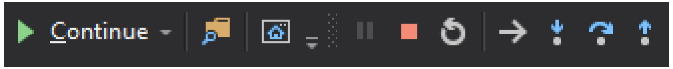


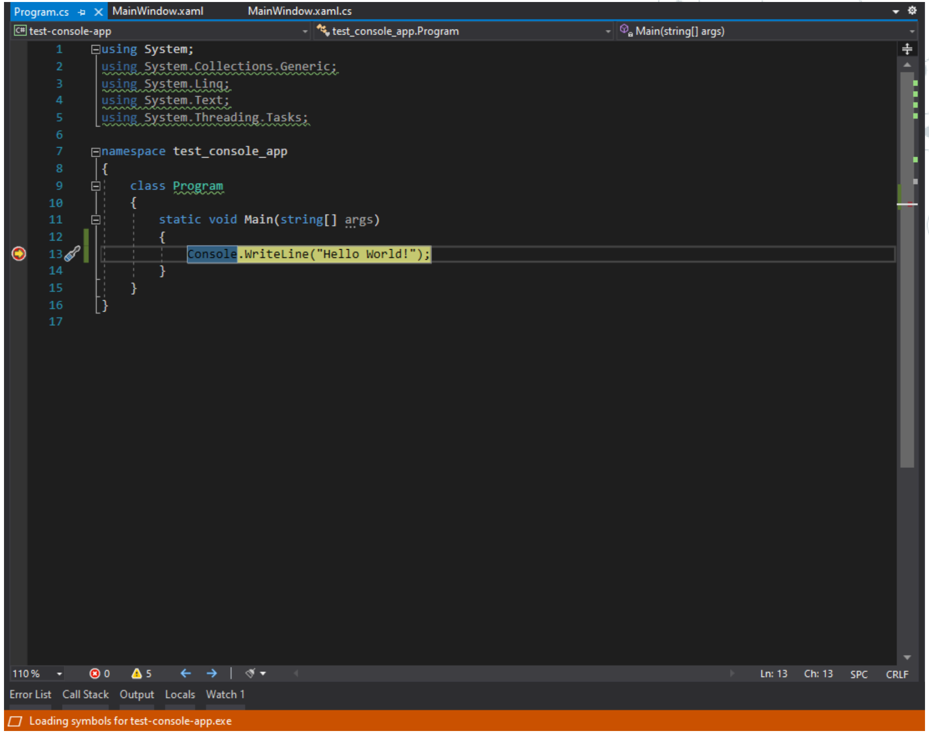
Show output from: Build

```
1>------ Rebuild All started: Project: wpf-net, Configuration: Debug Any CPU ------
1>  wpf-net -> D:\MPT\Document\Projects\wpf-net\wpf-net\wpf-net\bin\Debug\wpf-net.exe
========== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==========
```

Debugging

- Debug ➪ Start Debugging
- Add Break point (F9)





```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

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

Loading symbols for test-console-app.exe

Review Questions

1. What are the important components of .Net Framework?
2. What is meant by Managed and Unmanaged code?
3. What is an Assembly? What are the different types of Assemblies?

Hands-on Exercise

Create, Build and Run project below:

- Console App (.NET Framework)
- WPF App (.NET Framework)
- ASP.NET Core Web Application

Try to add break point and debug

Thanks!

Any questions?

You can find me at:

tranminhphuoc@gmail.com