

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

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles, suggesting a hierarchical or central structure. The lines are thin and gray, connecting the nodes in a non-linear fashion.

1.

What is .NET Framework?

A decorative network diagram in the bottom-right corner, similar to the one in the top-left. It shows a cluster of nodes connected by lines, with some nodes being larger and more prominent than others, indicating a network structure.

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

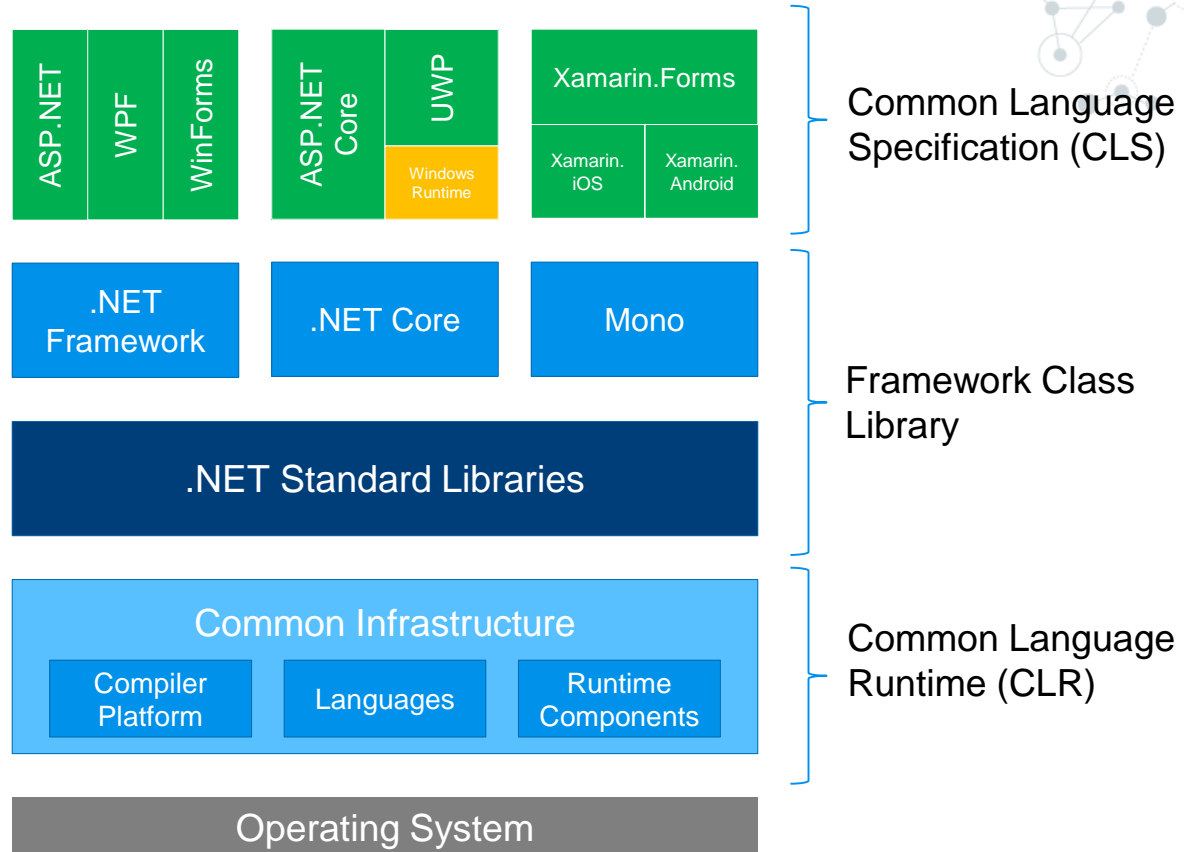
Developers use **the developer pack** (SDK – Software development kit) to create applications that run on .NET Framework.

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles inside, suggesting different levels of connectivity or importance. The lines are thin and gray, connecting the nodes in a non-linear fashion.

2. **Characteristics and architecture**

A decorative network diagram in the bottom-right corner, similar to the one in the top-left. It shows a cluster of nodes connected by lines, with some nodes being larger and more prominent than others. The overall structure is a complex, interconnected web.

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.



Architecture – Framework Class Library (FCL)

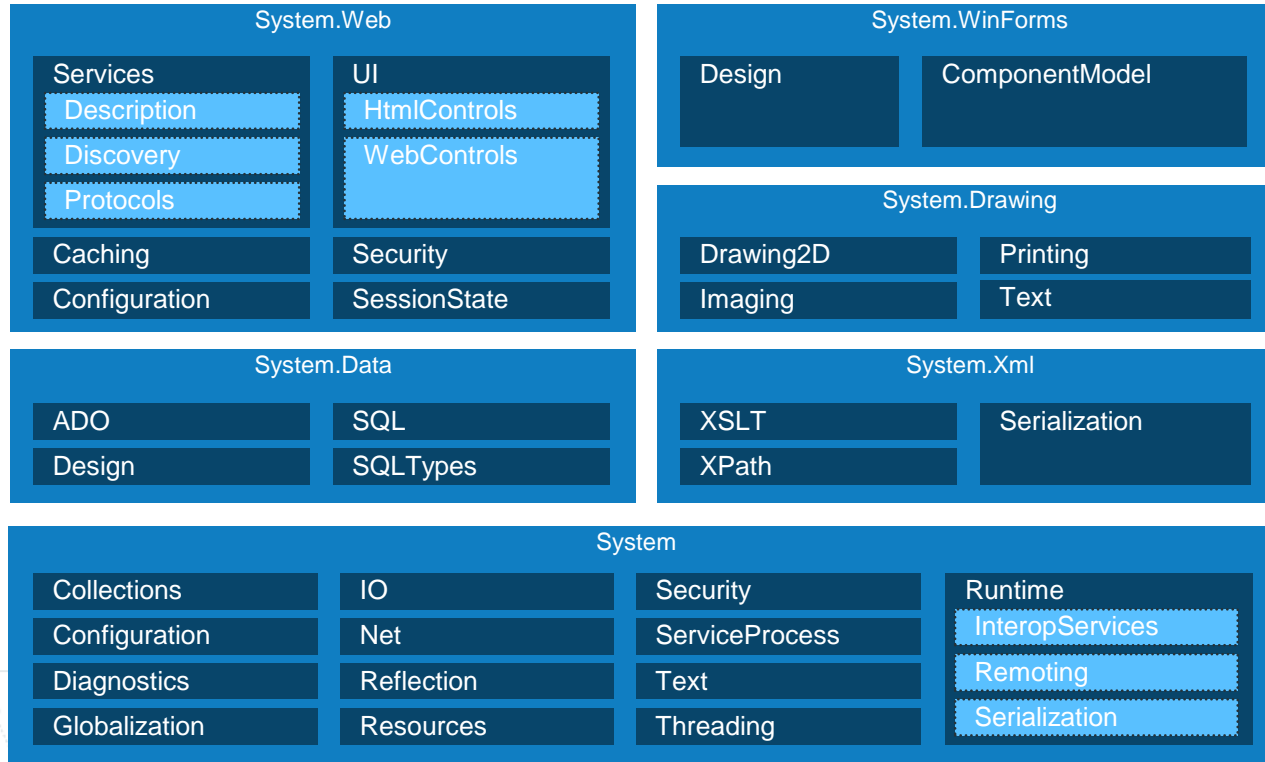
- ◎ The FCL provides the system functionality in the .NET Framework as it has various classes, data types, interfaces, etc. to perform multiple functions and build different types of applications such as desktop applications, web applications, mobile applications, etc.

.NET
Framework

.NET Core

Mono

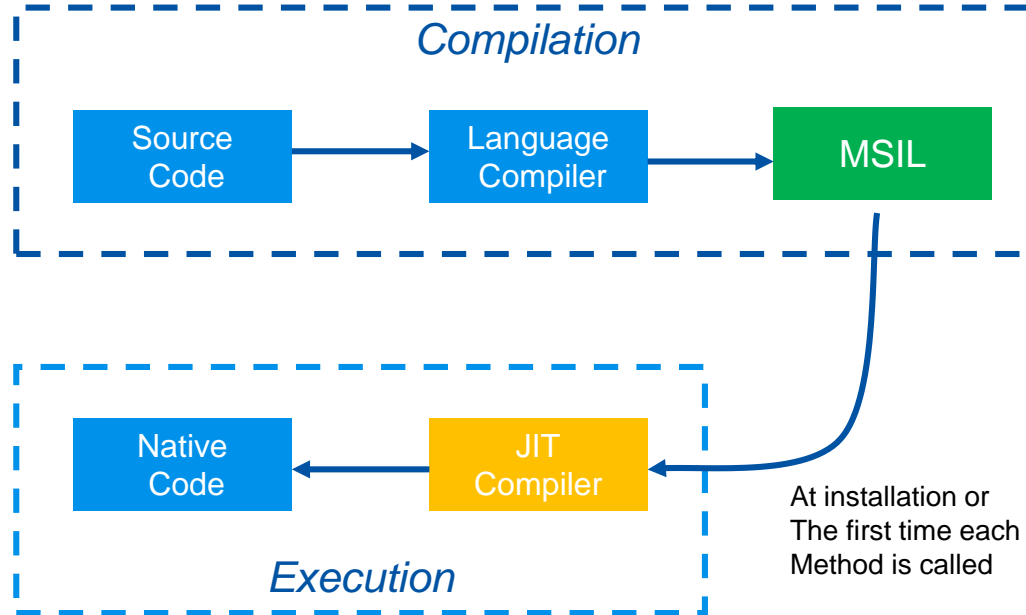
Architecture – Framework Class Library (FCL)



Architecture – Common Language Runtime (CLR)

- ◎ The CLR is the execution 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



Microsoft Intermediate Language (MSIL)

CPU independent set of instruction.

Can be efficiently converted to native code.

Just-in-time compiler (JIT)

Compiled at runtime

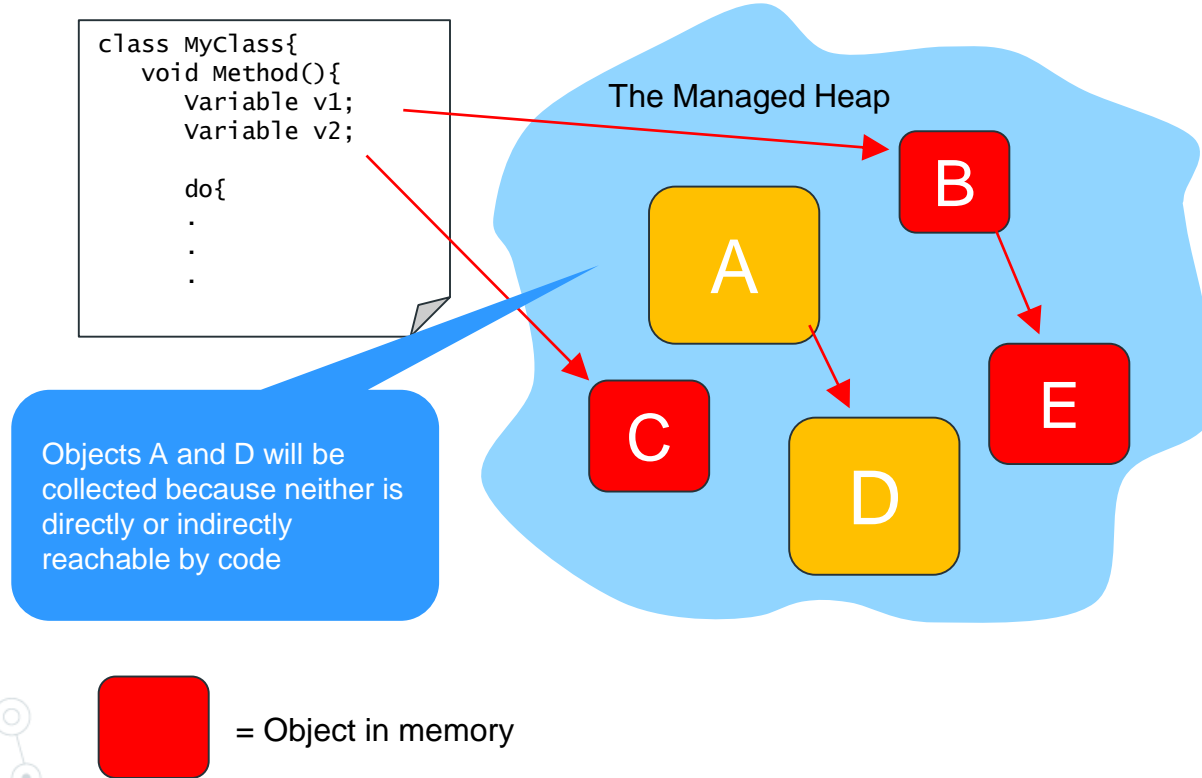
Convert MSIL to native code

At installation or
The first time each
Method is called

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



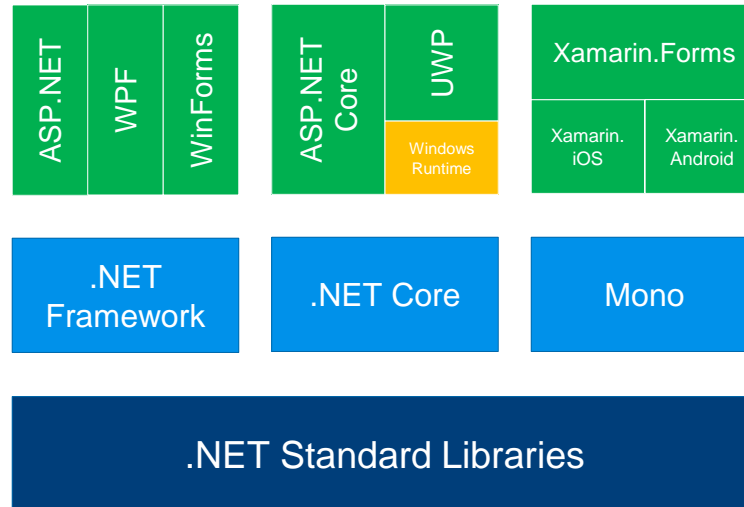
A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles, suggesting a hierarchical or central structure. The lines are thin and gray, connecting the nodes in a non-linear fashion.

3.

.NET Framework and .NET Core

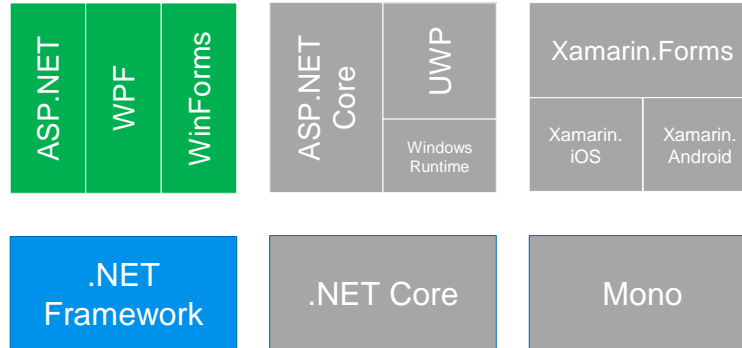
A decorative network diagram in the bottom-right corner, similar to the one in the top-left. It shows a cluster of nodes connected by lines, with some nodes being larger and more prominent than others, creating a sense of depth and connectivity.

.NET Framework and .NET Core



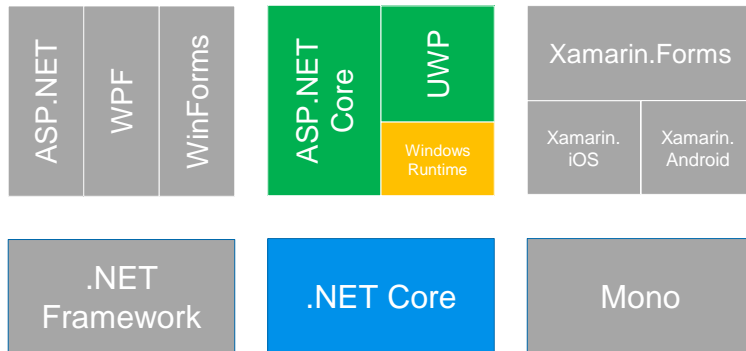
.NET Framework

- 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



.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

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles, suggesting different levels of connectivity or importance. The lines are thin and gray, creating a mesh-like structure.

4.

Using Visual Studio

A decorative network diagram in the bottom-right corner, similar to the one in the top-left. It shows a cluster of nodes connected by lines, with some nodes being larger and more prominent than others. The overall style is minimalist and technical.

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

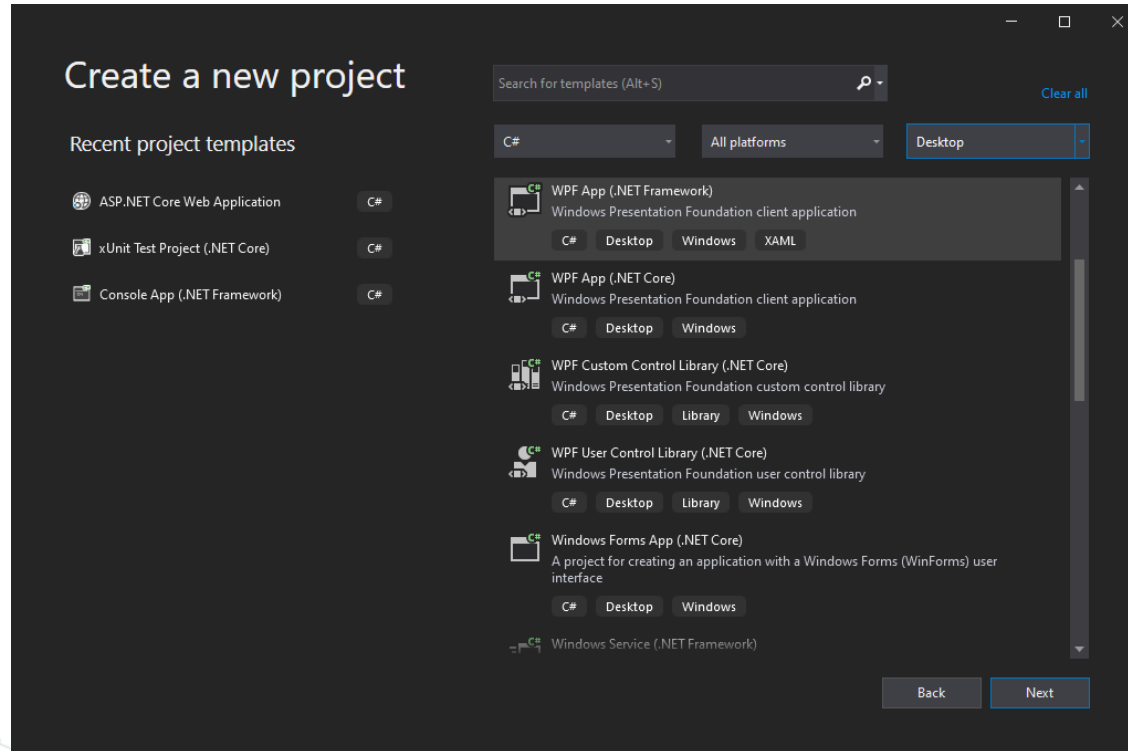
Hands-on Exercise

Download and install following software:

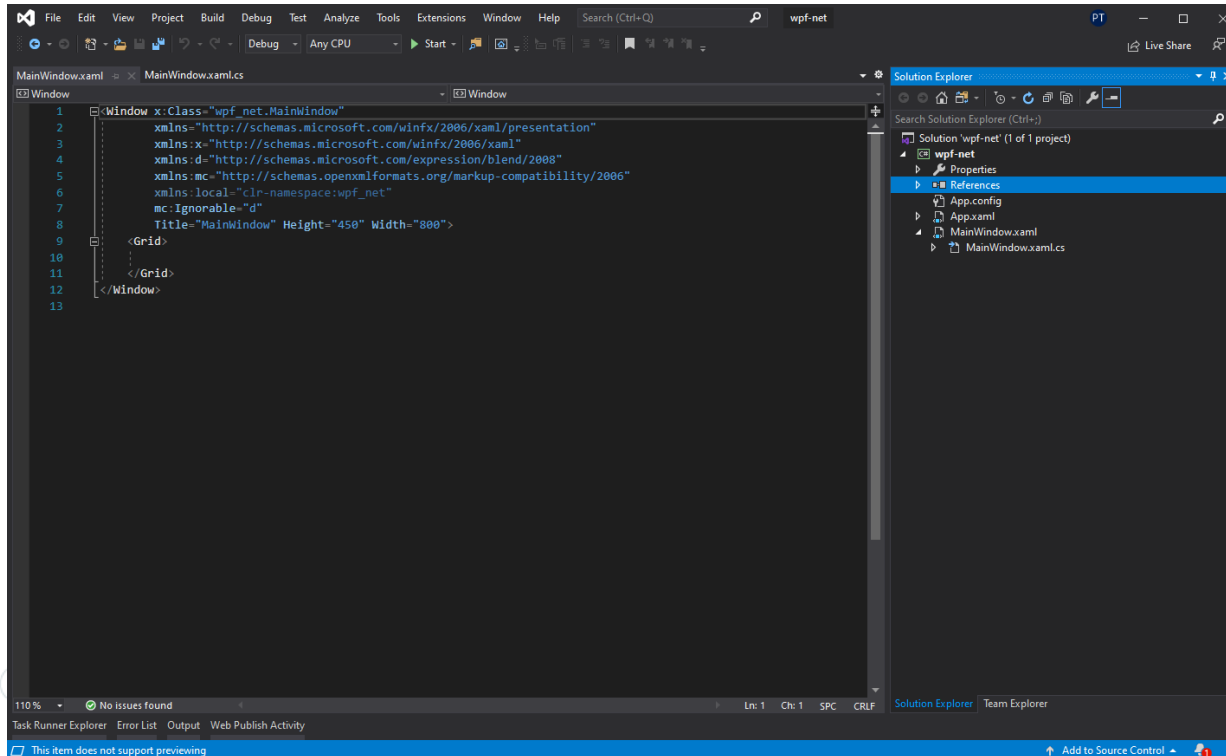
- Visual Studio 2019 (Community)
 - SQL Server 2019 Express



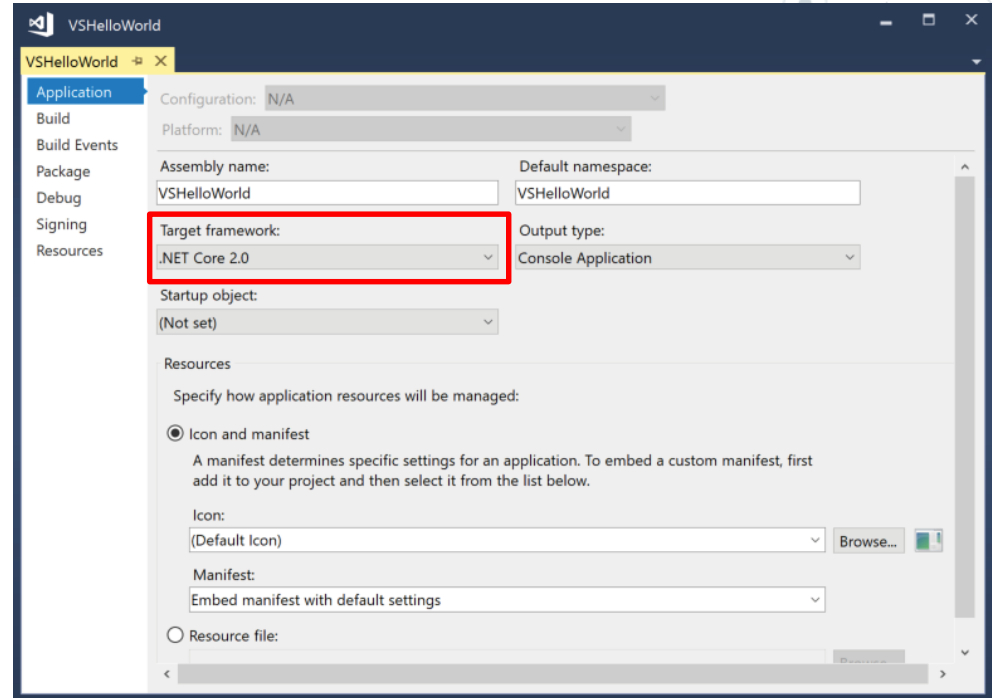
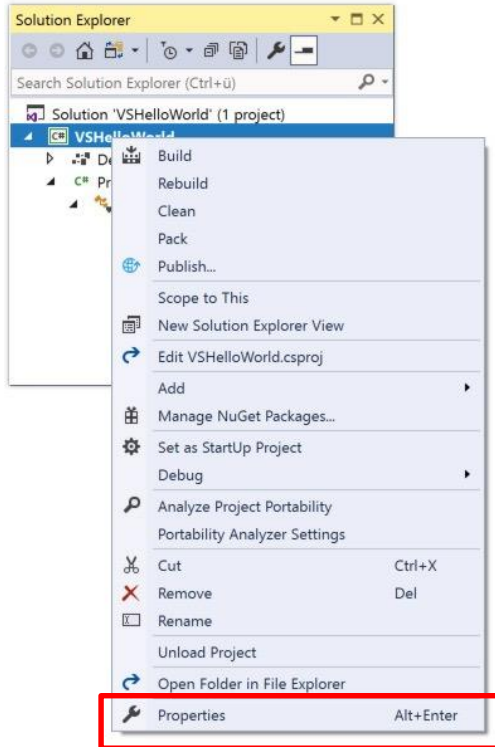
Creating a Project



Working with a Project

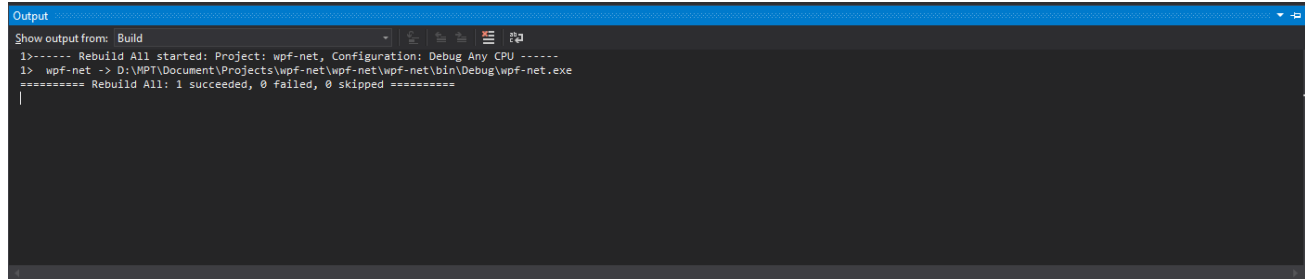


Configuring Project Properties



Building and running a Project

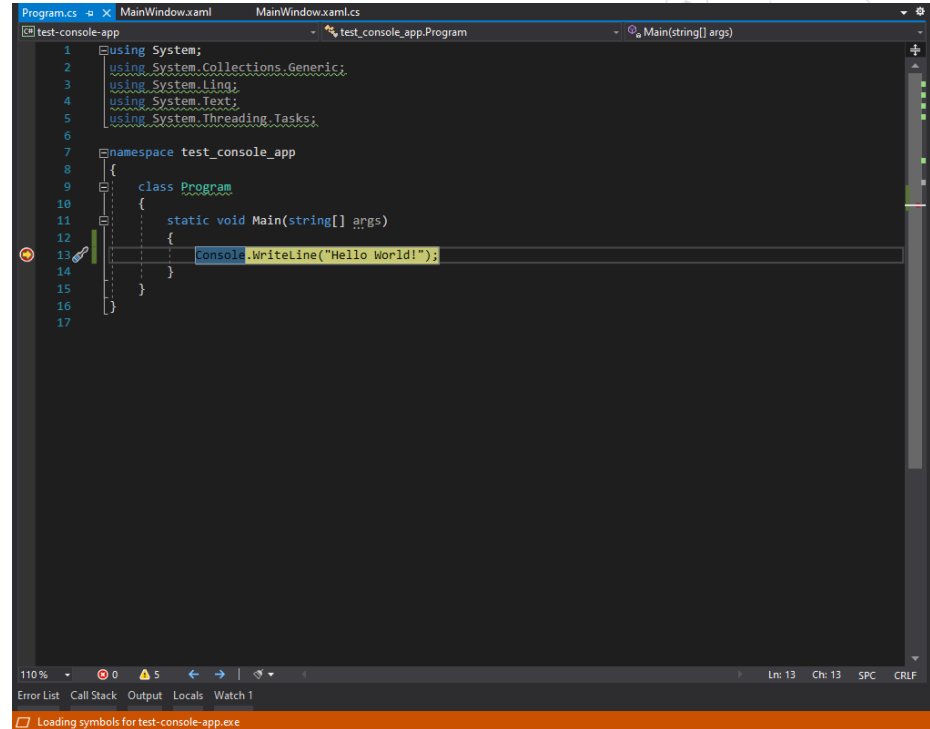
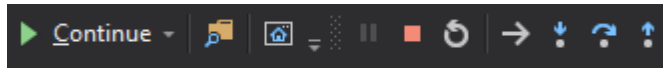
- ◎ Build ⇔ Build Solution
- ◎ Debug ⇔ Start Without Debugging



```
Output
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\bin\Debug\wpf-net.exe
===== Rebuild All: 1 succeeded, 0 failed, 0 skipped =====
```

Debugging

- ⦿ Debug ⇨ Start Debugging
- ⦿ Add Break point (F9)



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