

2021/2022(2)
IF184605 Framework-Based Programming

Lecture #3c

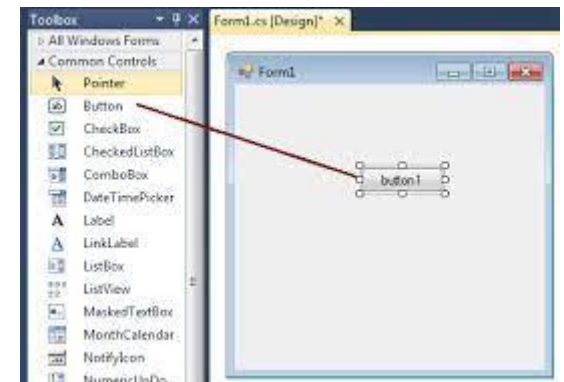
WinForms, WPF & UWP

Misbakhul Munir **IRFAN SUBAKTI**

司馬伊凡

Мисбакхул Мунир **Ирфан Субакти**

WinForms → WPF → UWP: History



- When .NET was released in 2000, you could use **WinForms** to make **desktop applications** for Windows. It provides access to 'native' controls (i.e., a Button made in WinForms is a Windows Button control from user32.dll) by wrapping the Windows API in managed code. These can run in any version of Windows which has the appropriate .NET framework version installed, including up to Windows 10.
- In 2006, **WPF** was released which is **an alternative to WinForms**. WPF uses **XAML**, which is a language based on **XML**, to declare the user interface elements. In a simple WPF app, the .xaml file describes the **GUI** and the code-behind file describes the **logic**. DirectX is used to draw whatever is described in the .xaml into the window.
- Like **WinForms**, programs made using **WPF can run in any version of Windows** which has the appropriate .NET framework version installed, including up to Windows 10. Note that using XAML is completely **optional**. Whatever user interface elements you describe using XAML can also be made using your .NET language **C#** or **VisualBasic.NET**. But using **XAML** is usually **simpler**.



WinForms → WPF → UWP: History (cont'd)

- From 2012 along with **Windows 8**, a new kind of app called metro/modern/windows-8/windows-store-apps can be made. These apps are **similar to WPF** in that they use **XAML for describing the GUI**. These apps can run only in **Windows 8, 8.1** and **Windows 10**. Windows 8.1 added some improvements and features but remained mostly the same. The same technology was used to make apps for **Windows Phone**.
- With the release of Windows 10, Microsoft made **UWP** (Universal Windows Platform) through which you can use the same code base to target different kinds of devices (desktop/mobile/Xbox/IoT/holographic). These **apps** made for the **UWP are similar to the earlier Windows 8/8.1 apps**. Like before, **XAML** is used to describe the **GUI**.
- Only for the **Windows Store apps**, there is **no backward compatibility**. The timeline goes **Windows 8 -> Windows 8.1 -> Windows 10/UWP**. An app made in UWP can only run in Windows 10 and an app made in Windows 8.1 can not run in Windows 8.

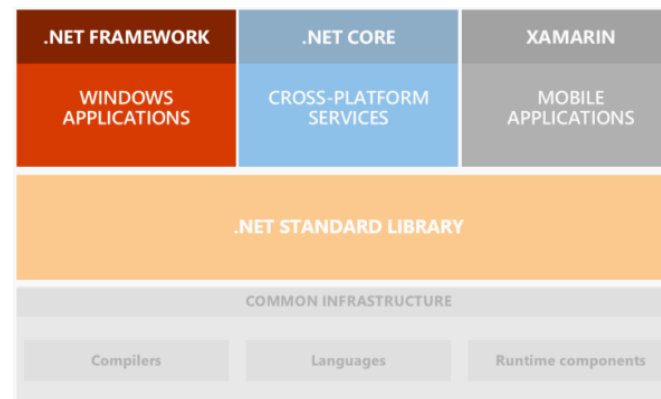


WPF: Definition

- WPF (Windows Presentation Foundation) is a UI framework that creates desktop client applications.
- The WPF development platform supports a broad set of application development features, including an application model, resources, controls, graphics, layout, data binding, documents, and security.



WPF: More about



- WPF is a technology released by Microsoft together with .NET Framework 3.0 in 2006.
 - In addition to WPF .NET 3.0 also includes several new technologies, namely WF (Workflow Foundation) for modelling and WCF (Windows Communication Foundation) which is the development of Web Services.
- WPF is a next-generation graphic platform that allows you to create an advanced UI (User Interface) that combines documents, media, 2D and 3D, and animations that are similar to web programming.
 - WPF can be used on Windows platforms such as Win XP, Vista, and Windows Server 2003/2008.

WPF: Background



- Before WPF technology, to develop Windows applications you had to use several different technologies.
 - E.g., to add user control forms such as buttons into your application you had to use the **Windows Forms (WinForms)** component.
 - Also, to add 2D Graphics you had to use the GDI+ library, for 3D Graphics you had to use the library DirectX or OpenGL.
- WPF is designed to provide a complete solution for application development.
 - WPF provides integration of different technologies
 - With WPF you can incorporate vector graphics, complex animation, and media components into applications.
 - A new feature in WPF (.NET 3.5) is the built-in 3D Graphics feature based on DirectX technology.

WPF is already old: Still relevant?

- Over the years it got improved and it is still now in the market.



WPF: What replaced it?

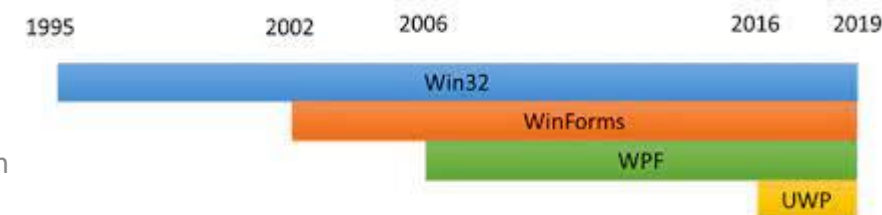
- Universal Windows Platform (UWP)
 - Both **WinForms** and **WPF** are old, and Microsoft is pointing developers towards its Universal Windows Platform (UWP) instead.
 - UWP is an evolution of the new application platform introduced in Windows 8 in 2012.



Development in C#: WPF or UWP?



- **C#** is used with WPF and UWP. So, you have a choice between WPF and UWP.
- If you are developing software for a *particular customer*, then choose WPF. WPF is more towards **WinForms** in terms of available resources. Since you don't have to publish this app to the world, **WPF is a good option**.
- **WPF** allows you to use the **SQL Server** database installed on the *desktop*. So you can have your database in SQL Server.
- How about **UWP**? You have the choice of **SQLite** database. UWP apps are distributed to the *world* using Windows Store. If you are planning software that can be used by *everyone* (no specific user/customer), then develop it using UWP and publish it via *windows store*.
- So? You can decide what to choose by asking yourself, what will my **customer/user base do with the software?**



Reference

- .NET Framework vs .NET Core vs .NET vs .NET Standard vs C#: link in [here](#).
- WinForm vs WPF vs UWP vs Console - The C# Desktop UI Showdown (and the future with .NET 5): link in [here](#).
- Creating WPF application in C#: WPF UI Programming (C#)
 - C# WPF UI Tutorials: 01 – The Basics: link in [here](#).
 - C# WPF UI Tutorials: 02 – TreeViews and Value Converters: link in [here](#).
 - C# WPF UI Tutorials: 03 – View Model MVVM Basics: link in [here](#).
 - C# WPF UI Tutorials: 04 – Custom Window Chrome and Styles: link in [here](#).
 - C# WPF UI Tutorials: 05 – Creating Login Form Sign Up Screen: link in [here](#).
 - C# WPF UI Tutorials: 06 – Attached Properties: link in [here](#).
 - C# WPF UI Tutorials: 07 – Storyboard Animations: link in [here](#).
 - .. and so on can be found in the playlist: link in [here](#).
- XAML Basics Tutorial | WPF: link in [here](#).
- Intro to UWP Apps in C#: link in [here](#).