**UWP (I have written all the programs in UWP)**

Universal Windows Platform (UWP) is a computing platform created by Microsoft and first introduced in Windows 10. The purpose of this platform is to help develop universal apps that run on Windows 10, Windows 10 Mobile, Windows 11, Xbox One, Xbox Series X/S and HoloLens without the need to be rewritten for each. It supports Windows app development using C++, C#, VB.NET, and XAML.

For creating Windows apps, the technology of choice should be the Universal Windows Platform. Of course, there are restrictions when this option is not available—for example, if you still need to support older O/S. versions like Windows 7. In this case you can use Windows Presentation Foundation (WPF).

**Developing apps with the Universal Windows Platform (UWP) Compared to WPF:**

UWP offers a more modern XAML to create the user interface. For example, data binding offers a compiled binding variant where you get errors at compile time instead of not showing the bound data. The application is compiled to native code before it’s run on the client systems. And it offers a modern design, which is now called Fluent Design from Microsoft.

In Universal Windows Apps would run on the mobile phones as well.

Universal Windows Platform (UWP) is a computing platform created by Microsoft and first introduced in Windows 10. The purpose of this platform is to help develop universal apps that run on Windows 10, Windows 10 Mobile, Windows 11, Xbox One, Xbox Series X/S and HoloLens without the need to be rewritten for each. It supports Windows app development using C++, C#, VB.NET, and XAML.

UWP not only can call the WinRT APIs that are common to all devices, but also APIs (including Win32 and .NET APIs) that are specific to the device family that the app is running on.

**Devices Supported by Windows 10**

Windows 8.1 and Windows Phone 8.1 apps target an OS; either Windows or Windows Phone. Windows 10 applications do not target an OS but they target one or more device families.

Device families have their own APIs as well, which add functionality for that particular device family. You can easily determine all the devices, within a device family, on which your applications can be installed and run from the Windows Store. Here is the hierarchical representation of the device family.

**Advantages of UWP**

Universal Windows Platform (UWP) provides a handful of things for developers. They are −

* One Operating System and One Unified Core for all the devices.
* One App Platform to run the applications across every family.
* One Dev Center to submit application and dashboard.
* One Store for all the devices.

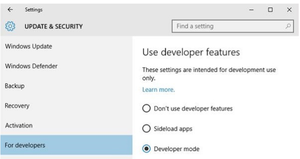
**Setup for UWP Development**

The following steps need to be followed to start creating your own Universal Windows Platform (UWP) apps for Windows 10.

Windows 10 OS − UWP apps need the latest version of Windows to develop. You can also develop UWP applications on Windows 8.1 but there is no support for UI designer Window.

Windows 10 developer tools − In Visual studio 2015, you can design, code, test, and debug your UWP apps. You can download and install the free Microsoft Visual Studio Community 2015 from https://dev.windows.com/en-us/downloads

* Enable development mode for Windows 10 −
* Go to Start > Settings.
* Select Update & security.
* Then select "For developers".



WinUI



* Windows UI Library (WinUI codenamed "Jupiter", and also known as UWP XAML and WinRT XAML) is a user interface API that is part of the Windows Runtime programming model that forms the backbone of Universal Windows Platform apps (formerly known as Metro-style or Immersive) for the Windows 8, Windows 8.1, Windows 10 and Windows Phone 8.1 operating systems.
* It enables declaring user interfaces using Extensible Application Markup Language (XAML) technology.
* **WinUI is a user interface layer that contains modern controls and styles for building Windows apps.**As the native UI layer in Windows it embodies Fluent Design, giving each Windows app the polished feel that customers expect.
* WinUI 2 is a library of controls and styles currently available for use in any UWP app. It offers exciting, flexible, modern controls such as NavigationView and TeachingTip.
* WinUI 3 is the next generation of the WinUI framework, and is now available for production apps. It dramatically expands WinUI into a full UX framework, and ships as a part of the [Windows App SDK](https://docs.microsoft.com/windows/apps/windows-app-sdk/) . With WinUI 3, modern controls, styles, and capabilities will be available to apps on Windows 11 and Windows 10.

**Who can use WinUI?**

WinUI 2 is compatible with UWP apps, and WinUI 3 works with any app supported by the Windows App SDK. WinUI 3 can be used *directly* as the UI layer for desktop apps, or starting next year, it can be used to modernize a Win32 app's UI *gradually*, using [XAML Islands](https://docs.microsoft.com/windows/apps/desktop/modernize/xaml-islands) to mix and match with the following technologies:

* WPF
* WinForms
* MFC
* ComCtl32

WinUI also serves as the basis for cross-platform technologies that provide great native Windows experiences. [React Native for Windows](https://microsoft.github.io/react-native-windows/), [.NET MAUI](https://devblogs.microsoft.com/dotnet/announcing-net-maui-preview-5/), and the [Uno platform](https://platform.uno/) harness the power of WinUI today - if you're currently using those platforms you're already using WinUI!