Universal Windows Platform 编程学习笔记

Roger Young

2017 年 12 月 8 日

目录

第一章 Get started with the Universal Windows Platform

1.1 What's a Universal Windows Platform (UWP) app?

The Universal Windows Platform (UWP) is the app platform for Windows 10. You can develop apps for UWP with just one API set, one app package, and one store to reach all Windows 10 devices –PC, tablet, phone, Xbox, HoloLens, Surface Hub and more. It' s easier to support a number of screen sizes, and also a variety of interaction models, whether it be touch, mouse and keyboard, a game controller, or a pen. At the core of UWP apps is the idea that users want their experiences to be mobile across ALL their devices, and they want to use whatever device is most convenient or productive for the task at hand.

UWP is also flexible: you don't have to use C# and XAML if you don't want to. Do you like developing in Unity or MonoGame? Prefer JavaScript? Not a problem, use them all you want. Have a C++ desktop app that you want to extend with UWP features and sell in the store? That's okay, too.

The bottom line: You can spend your time working with familiar programming languages, frameworks and APIs, all in single project, and have the very same code run on the huge range of Windows hardware that exists today. Once you've written your UWP app, you can then publish it to the store for the world to see.

1.1.1 So, what exactly is a UWP app?

What makes a UWP app special? Here are some of the characteristics that make UWP apps on Windows 10 different.

- There's a common API surface across all devices.

  The Universal Windows Platform (UWP) core APIs are the same for all classes of Windows device. If your app uses only the core APIs, it will run on any Windows 10 device, no matter if you are targeting a desktop PC, an Xbox or a Mixed Reality headset.

- Extension SDKs let your app do cool stuff on specific device types.

  Extension SDKs add specialized APIs for each device class. For example, if your UWP app targets HoloLens, you can add HoloLens features in addition to the normal UWP core APIs. If you target the universal APIs, your app package can run on all devices that run Windows 10. But if you want your UWP app to take advantage of device specific APIs in the event it is running on a particular class of device, you can check at run-time if an API exists before calling it.

- Apps are packaged using the .AppX packaging format and distributed from the Store.

  All UWP apps are distributed as an AppX package. This provides a trustworthy installation mechanism and ensures that your apps can be deployed and updated seamlessly.

- There's one store for all devices.

  After you register as an app developer, you can submit your app to the store and make it available on all types device, or only those you choose. You submit and manage all your apps for Windows devices in one place.

- Apps support adaptive controls and input

  UI elements use effective pixels (see Responsive design 101 for UWP apps), so they can respond with a layout that works based on the number of screen pixels available on the device. And they work well with multiple types of input such as keyboard, mouse, touch, pen, and Xbox One controllers. If you need to further tailor your UI to a specific screen size or device, new layout panels and tooling help you adapt your UI to the devices your app may run on.

1.1.2 Use a language you already know

UWP apps use the Windows Runtime, a native API built into the operating system. This API is implemented in C++ and supported in C#, Visual Basic, C++, and JavaScript. Some options for writing apps in UWP include:

- XAML UI and a C#, VB, or C++ backend
- DirectX UI and a C++ backend
- JavaScript and HTML

Microsoft Visual Studio 2017 provides a UWP app template for each language that lets you create a single project for all devices. When your work is finished, you can produce an app package and submit it to the Windows Store from within Visual Studio to get your app out to customers on any Windows 10 device.

1.1.3 UWP apps come to life on Windows

On Windows, your app can deliver relevant, real-time info to your users and keep them coming back for more. In the modern app economy, your app has to be engaging to stay at the front of your users’ lives. Windows provides you with lots of resources to help keep your users returning to your app: