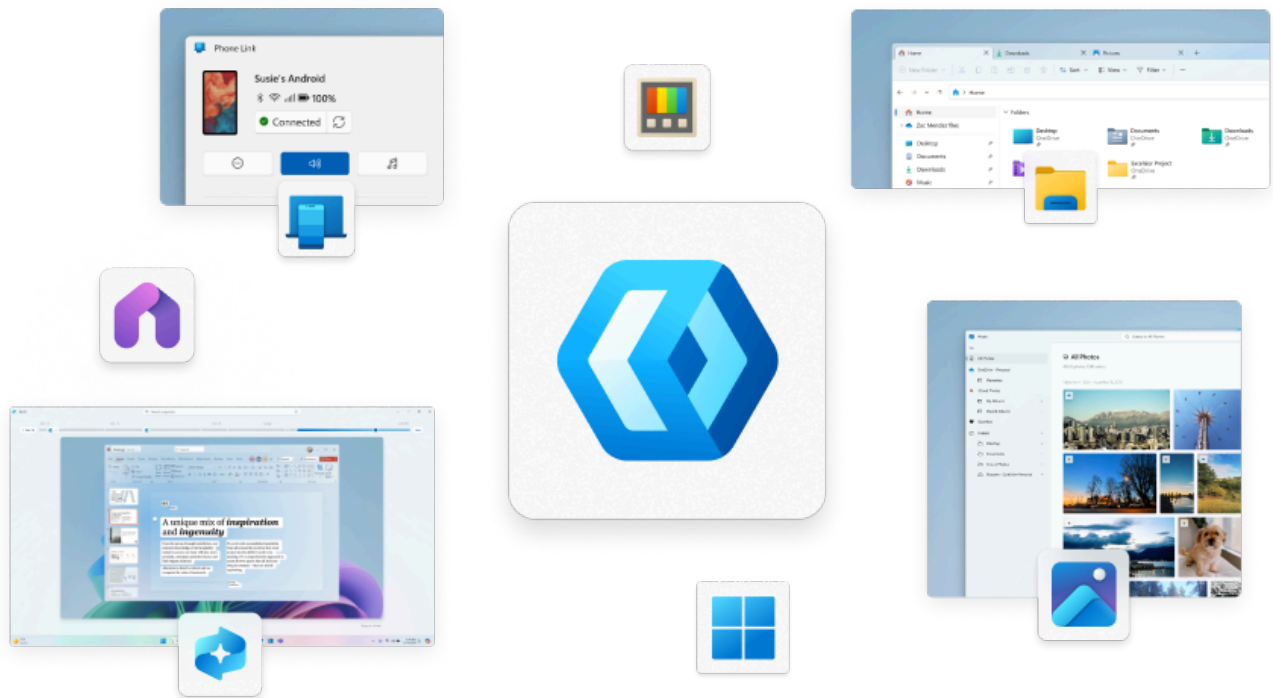


WinUI 3 | Ep1. 初识WinUI 3 & 环境配置



WinUI 3是微软为现代 Windows 应用开发而制作的最新的UI框架，包含在Windows App SDK中，也是Windows系统的组成部分。它来源于坟头草已经三尺高的UWP平台的UI控件库，WinUI 2，带有一套与Windows 11设计风格完全一致的控件，也是微软 Fluent Design System 设计语言的官方实现。

什么？又一个UI框架？

Windows 进入了它的第40个年头（Windows 1.0 于 1985年11月20日发布）。尽管 Windows 保持了惊人的向后兼容性，即便是在今天的 Windows 11 24H2 上，一些来自于几十年前Windows 3.1的老旧软件仍然可以运行。

作为一个一直保持活跃的系统，它总是要向前走的，不能只是停留在过去。在这漫长的时间里，Windows平台的软件开发经历了多次进化。微软自己的Windows原生应用开发框架从最早期Win32，到MFC，到WinForm、WPF，到Metro App、UWP，而后是发布于Windows 11公开前夕 2020年的Project Reunion，也就是Windows App SDK。

这些不同的开发框架可以共存于最新版本的 Windows 中，如何选择最合适的框架成了开发者的难题。尤其是在 Windows 10 时代，许多现代功能和特性（如原生触控支持、现代应用权限管理、MSIX 安装包与包管理、Windows 系统集成等）都是 UWP 特有的。UWP为了实现微软设想

的全平台运行的大一统愿景，而与老旧应用生态彻底隔绝关系，最终随着 Windows Phone 的退出舞台而逐渐消亡。

Project Reunion

UWP虽然失败了，微软也认识到了像UWP这样彻底重新开辟应用生态是错误的，但下一代Windows依旧需要一套应用开发框架。所以这个框架需要：

- 能轻松制作出符合Windows设计语言的界面。
- 既能让现有的老程序也能接入Windows的新特性，完成应用生态现代化。
- 也能继承UWP的遗产，因为Windows的相当多系统应用都是基于UWP的。

因此Project Reunion诞生了，它的使命就是完成整个Windows应用开发生态的统一，让无论是老旧的应用，还是新的应用，都能平等的享受Windows所提供的最新特性。UWP先进的UI也被分离出来，成为了WinUI 3。尽管桌面端开发已不再是微软发力的重点，但 WinUI 3 发展了几年，到今天已经比较成熟。从Windows 11 24H2开始，越来越多的系统组件也从UWP转向WinUI 3。

你说得对，但是为什么选择它？

- 它是现在和下一个大版本Windows的主要应用开发框架。
- 它有开箱即用的漂亮界面，能与Windows系统完美融合，并且原生支持各种Windows的现代特性。WinUI 3 的Composition层（也叫可视化层）将动画和特效与应用进程相分离，使其可以高性能的实现其它原生框架难以实现的华丽的效果。
- 即使你就是认为它太局限，没啥前景，但它与C#其它的UI框架高度相似，许多经验与知识都是相通的。

如果你只是想制作一个在Windows上运行的应用，那么 WinUI 3 就是你的最佳选择。

好了，废话不说了，那我们开始吧！

Quick Start | 第一个 WinUI 3 应用

在这篇教程中，我们将使用 Visual Studio 2022 与 .NET 9，使用C#语言进行开发。

作为原生UI框架，WinUI 3除了主推的C#，亦支持C++（官方支持）和Rust等语言进行开发。

如果你希望使用它们进行开发，本系列教程的一些内容可能对你帮助不大。

配置开发环境

首先，让我们从安装 Visual Studio 2022 开始。WinUI 3 应用在 Visual Studio 中拥有最佳的调试体验和功能支持。

安装 Visual Studio 2022

如果你已经安装了Visual Studio 2022，可以跳至下一小节。

除了到官网下载或者去Microsoft Store下载，对于Windows 11，你还可以通过Windows的包管理器 `winget` 快速下载最新版的VS 2022。打开Windows Terminal，输入以下命令并执行以安装正式版的VS 2022。

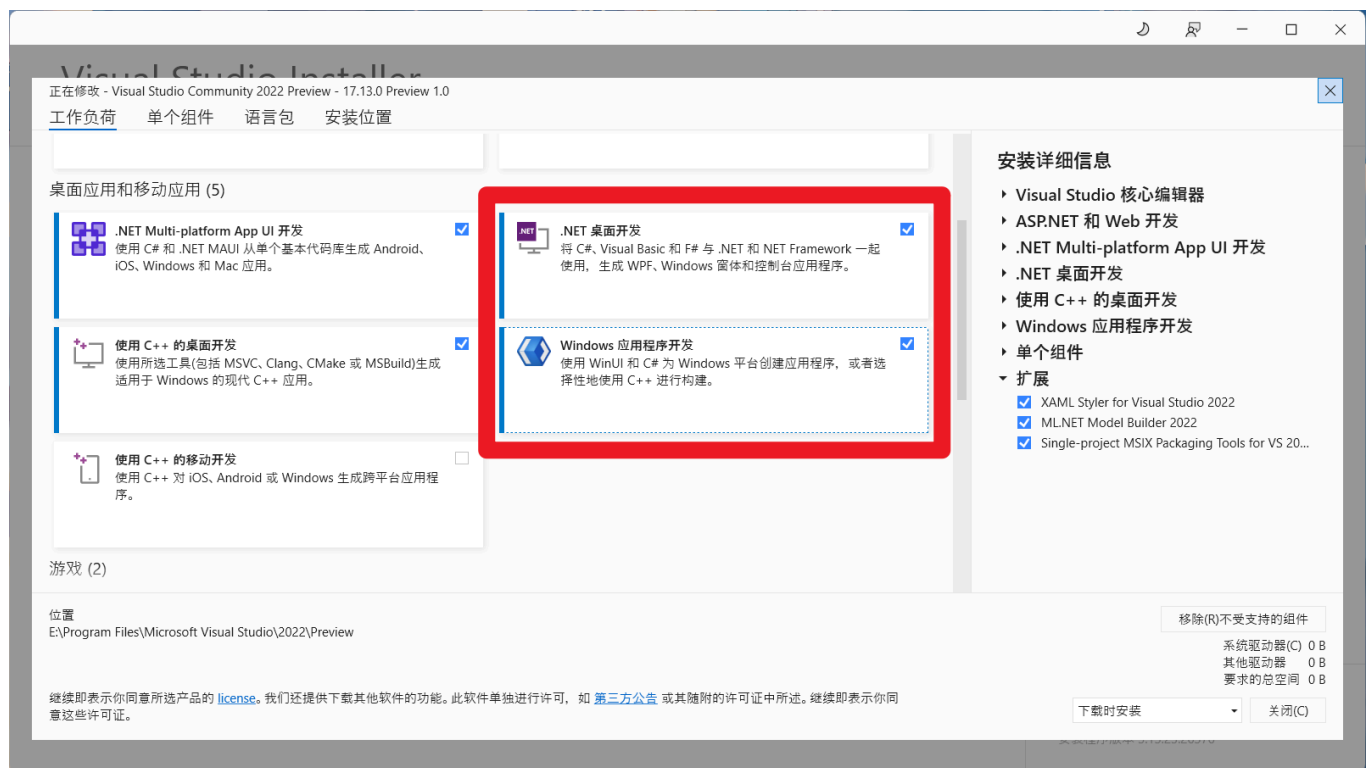
```
winget install Microsoft.VisualStudio.2022.Community
```

等待下载与安装结束，Visual Studio Installer 会自动打开。

安装工作负载

编辑你的VS 2022安装，确保勾选

- .NET 桌面开发
- Windows 应用程序开发



可选： 如果你的系统中没有安装Git for Windows，你可以在单个组件中找到“适用于Windows的Git”。在登陆微软账户与GitHub账号后，Visual Studio会自动为你完成Git相关配置。

启用 Windows 开发人员模式

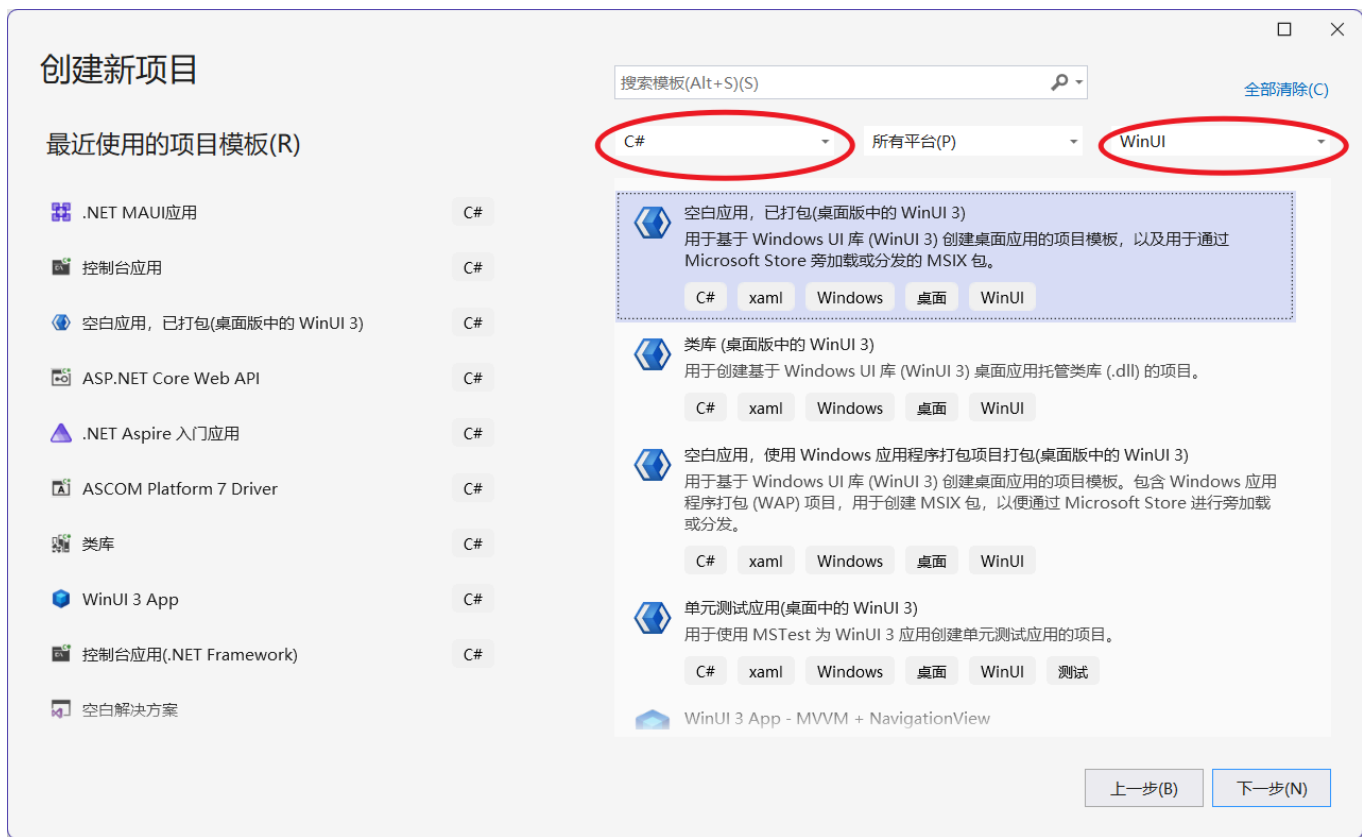
打开设置 > 系统 > 开发者选项，启用开发人员模式，按提示操作。

强烈建议将默认终端更换为 Windows Terminal，如果你还没有设置过的话。
还建议在此处启用允许本地未签名的 PowerShell 脚本执行。

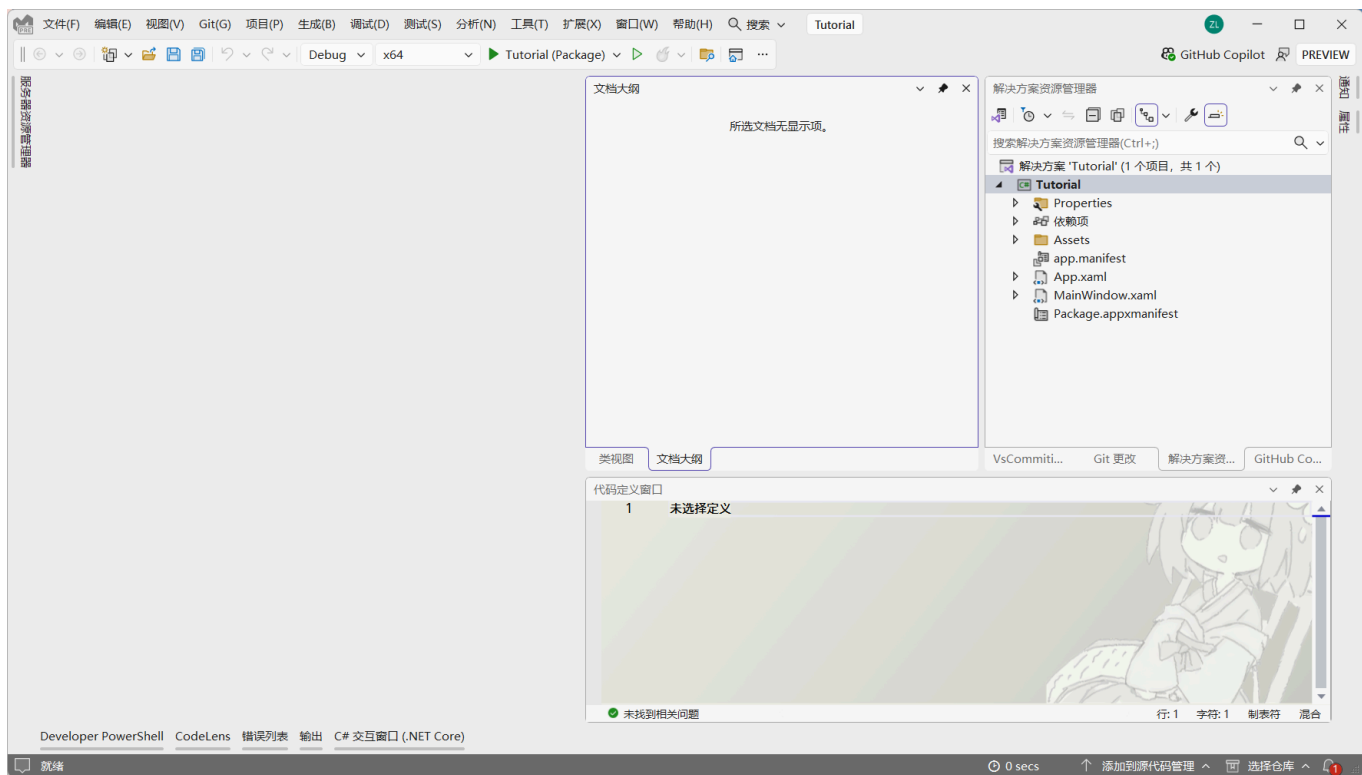


创建新项目

打开 Visual Studio 2022，创建一个新项目。选择“空白应用，已打包（桌面版中的 WinUI 3）”



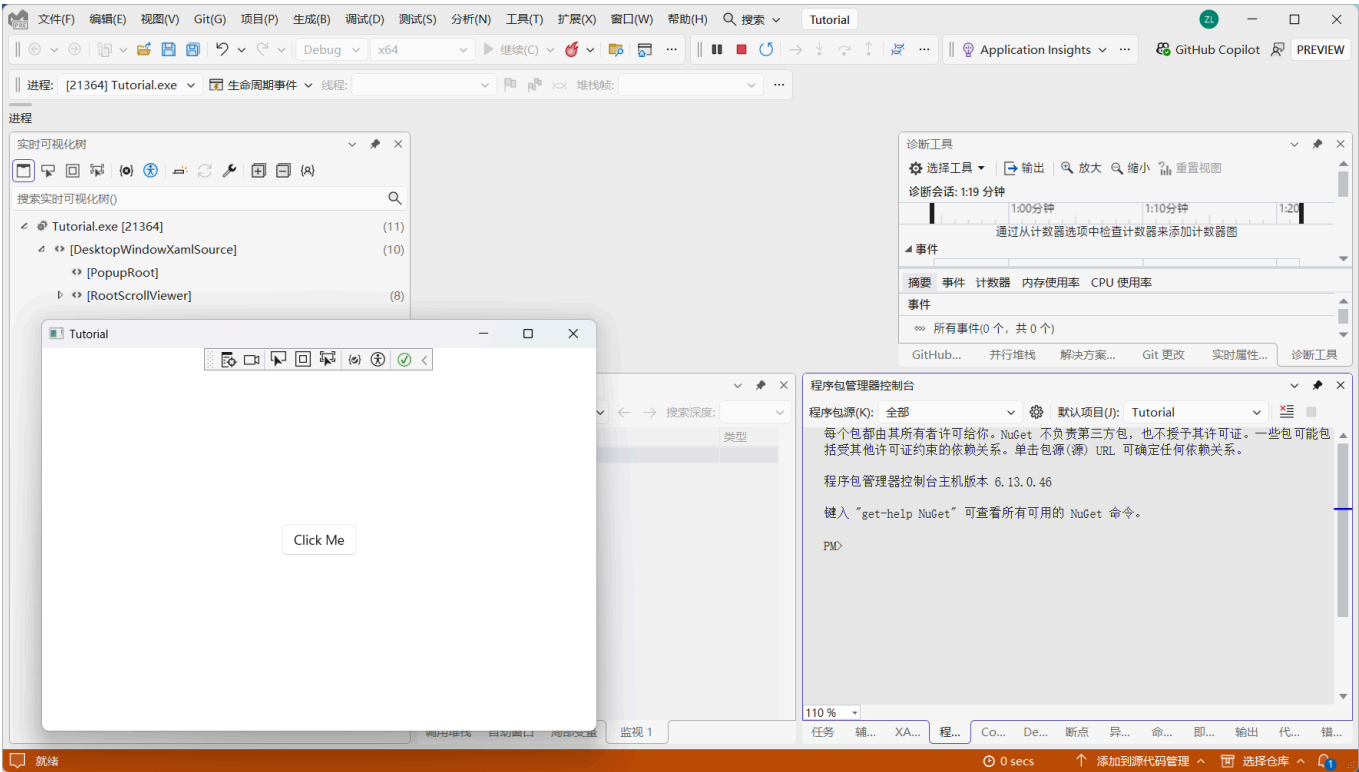
创建后，你应该看到如下界面：



你的界面布局可能与我的有所不同，但这并不影响实际操作。

试着运行吧！

确保在界面顶部，实心绿色三角形右边的启动项目选择的是“你的项目名（Packaged）”，而不是 Unpackaged 的版本。点击绿色三角形，Visual Studio 将开始编译并部署项目，启动调试。当你看到一个带有按钮的窗口时，说明你已成功完成了环境配置。恭喜你，迈出了开发的第一步！



试着点击按钮，你会发现按钮的内容发生了变化。如果你的系统启用了深色模式，应用也会原生支持深色主题，而无需额外编写代码。这也充分体现了 WinUI 3 天生就能充分利用 Windows 的众多新功能。