WinUI - Getting started

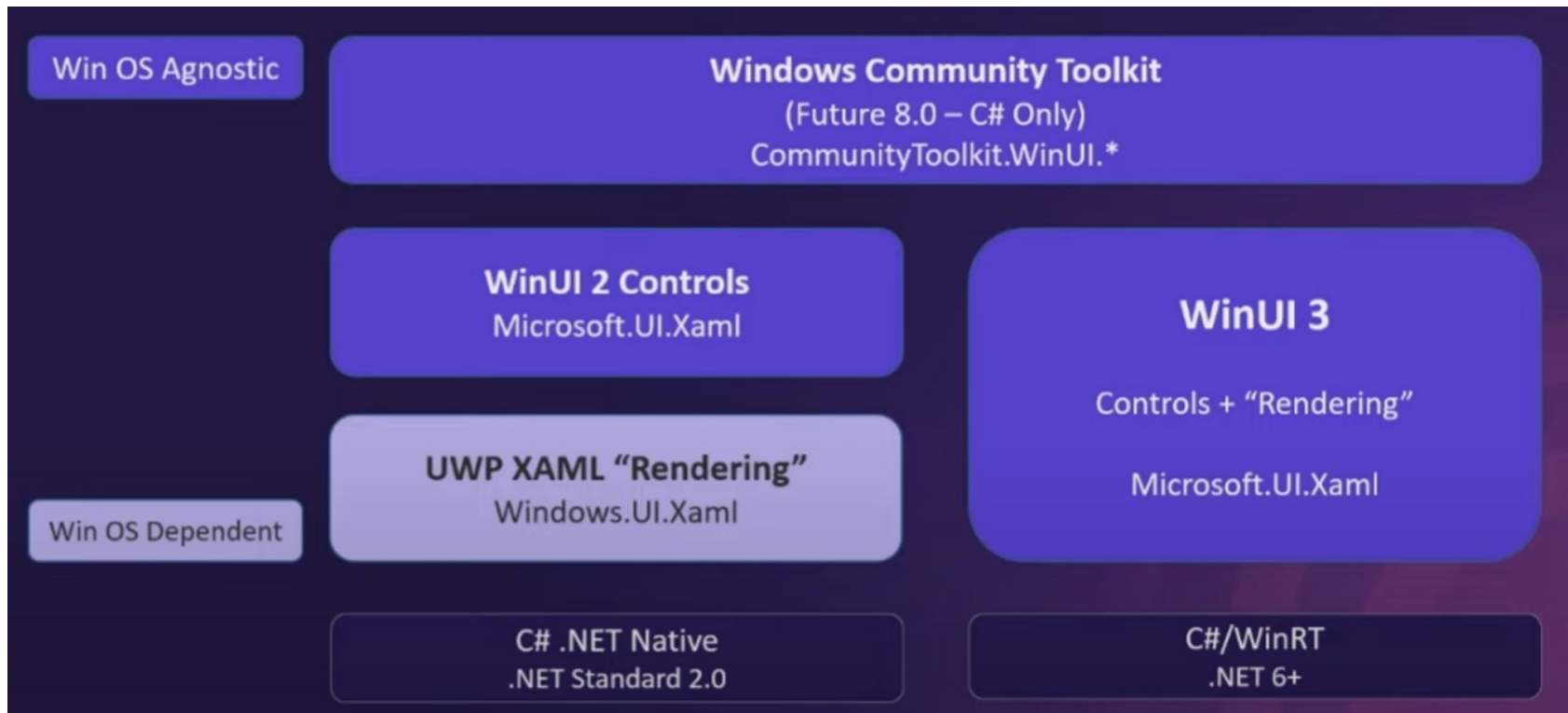https://learn.microsoft.com/en-us/windows/apps/get-started

What is WinUI?

- ❏ **XAML** markup-based user interface layer
- ❏ Contains **modern controls** and **styles** for building Windows apps
- ❏ **Native UI layer** for **Windows App SDK**
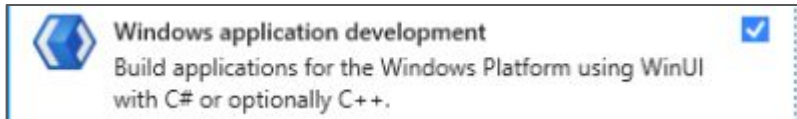   - ❏ Embodies **Fluent Design**
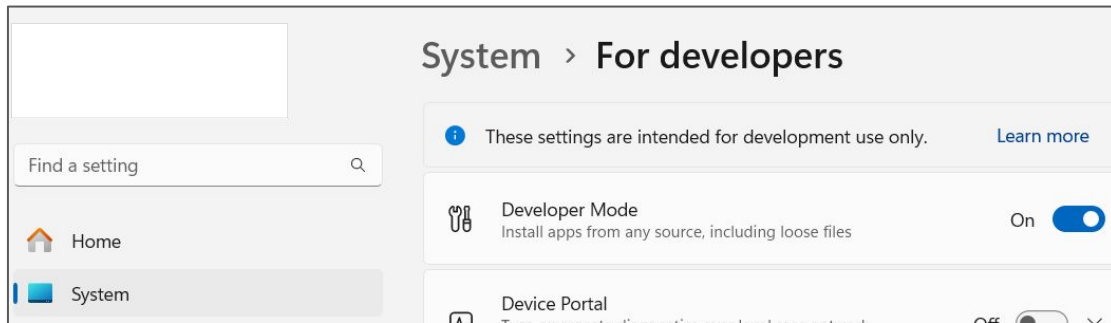
History

Evolution

4

Getting started

1. Install **Visual Studio** (Community Edition is enough)
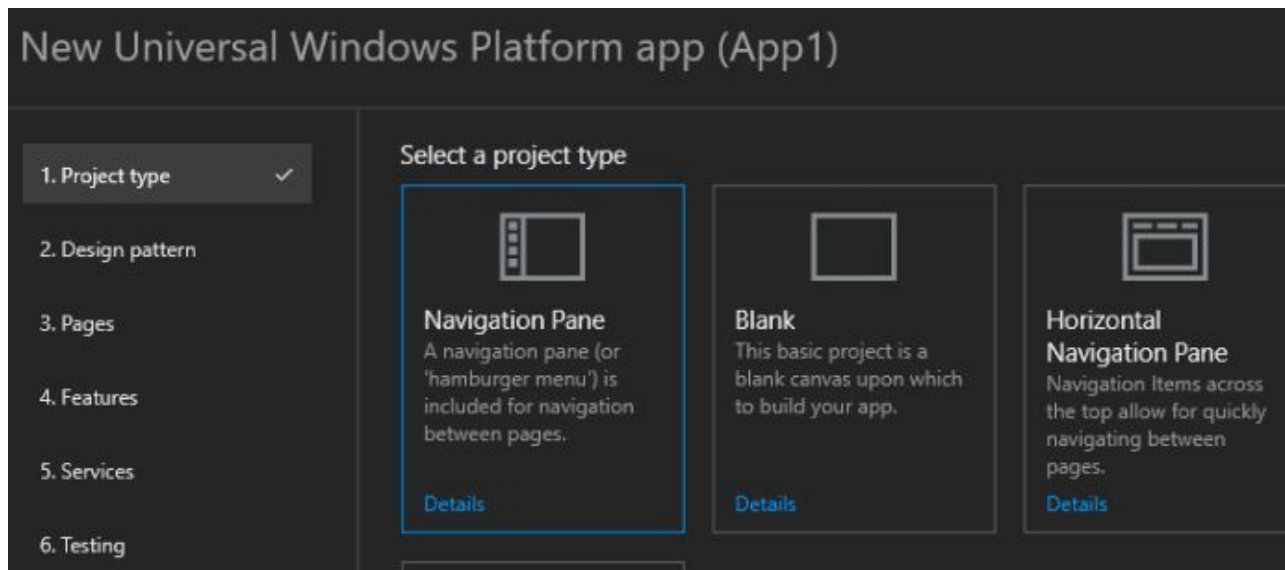   - ❏ Required workload:

2. Enable **Developer Mode**

5

3. Extension: Template Studio for WinUI

❏ Create apps using C# and the MVVM Toolkit

    ❏ **Must use** the **MVVM** design pattern

❏ Support multiple layout

Create your first WinUI app

Build & Launch the app: Ctrl + Shift + B > Ctrl + F5
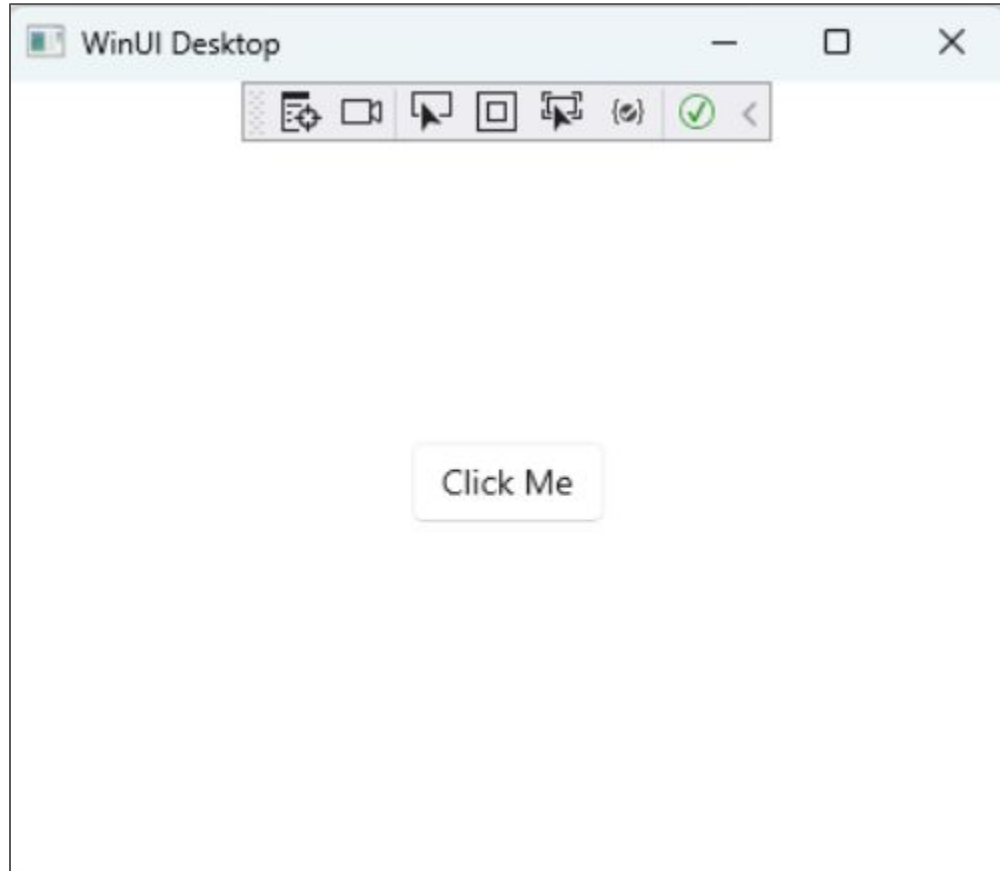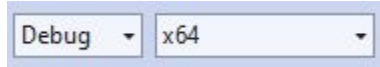
Project structure & Artifacts
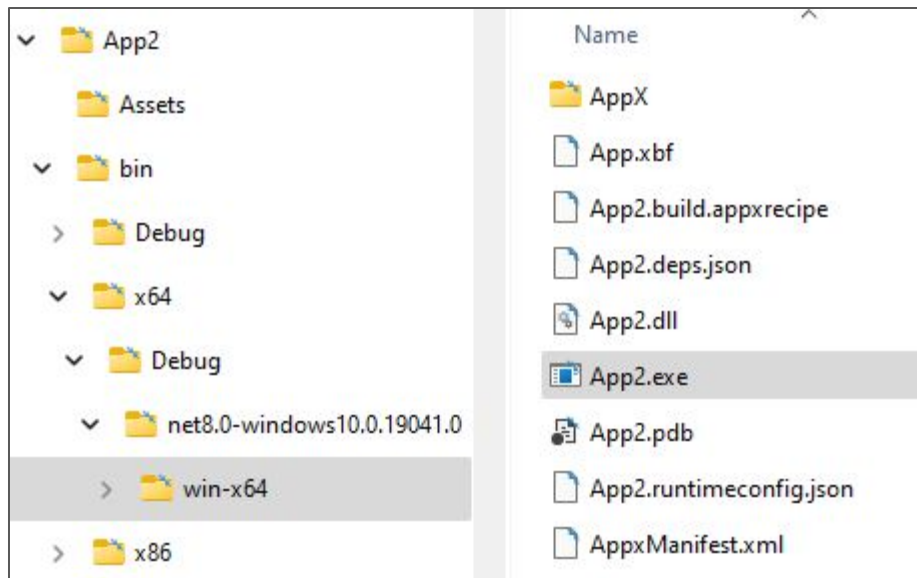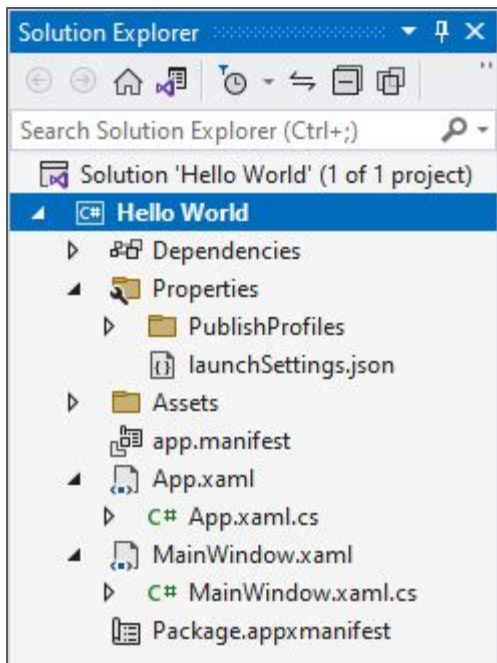
Currently you **cannot** run this exe file directly!
You **must deploy it**!

UI definition: MainWindow.xaml

```xml
<Window
    x:Class="App1.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:App1"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d">

    <StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Vertic
        <Button x:Name="myButton" Click="myButton_Click">Click Me</Button>
    </StackPanel>
</Window>
```

XML Quick summary

```xml
<?xml version="1.0" encoding="utf-8"?>
<Root>
    <Student>
        <Name>Tran Duy Quang</Name>
        <Email>tdquang@gmail.com"</Email>
    </Student>
    <Student Name="Nguyen Dieu Linh" Email="ndlinh@gmail.com" />

    <Class Name="Windows programming">
        <Student Name="Le Thai Minh"
            Class.Role="Monitor" />
    </Class>
</Root>
```

Event handler mapping: MainWindow.xaml.cs

```xml
<Button x:Name="myButton"
        Click="myButton_Click">
    Click Me
</Button>
```

```csharp
private void myButton_Click(
    object sender, RoutedEventArgs e)
{

    myButton.Content = "Clicked";

}
```

Entry point: App.xaml.cs > OnLaunched

```csharp
protected override void OnLaunched(
    Microsoft.UI.Xaml.LaunchActivatedEventArgs args)
{

    m_window = new MainWindow();
    m_window.Activate();
}

private Window m_window;
```

Minor changes

- Create a red "Hello world" text instead of a button
- Change the **Title** to "Hello world"
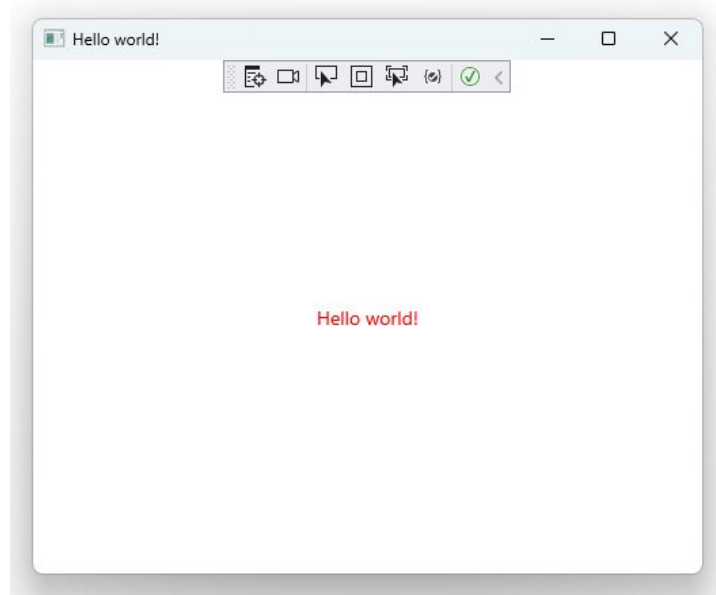
Another change
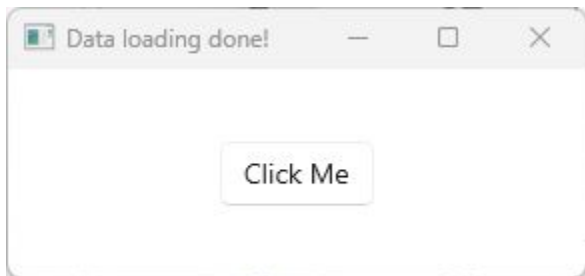

```
public MainWindow()
{
    this.InitializeComponent();
    this.Activated += MainWindow_Activated;
}


1 reference
private void MainWindow_Activated(
    object sender, WindowActivatedEventArgs args)
{
    this.Title = "Data loading done!";
}
```

Clean for submission

- Build > Clean solution
- Delete these folders: .vs, bin, obj

Your compressed file should be very small!

Local Publish & Deploy
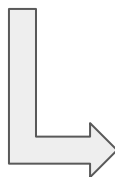
Create msix file by creating a package

New Solution Explorer View

File Nesting ▸

Edit Project File

Add ▸

Package and Publish ▸

Manage NuGet Packages...

Manage User Secrets

Remove Unused References...

Sync Namespaces

Configure Startup Projects

Associate App with the Store...
Convert Content Group Map File
Create App Packages...
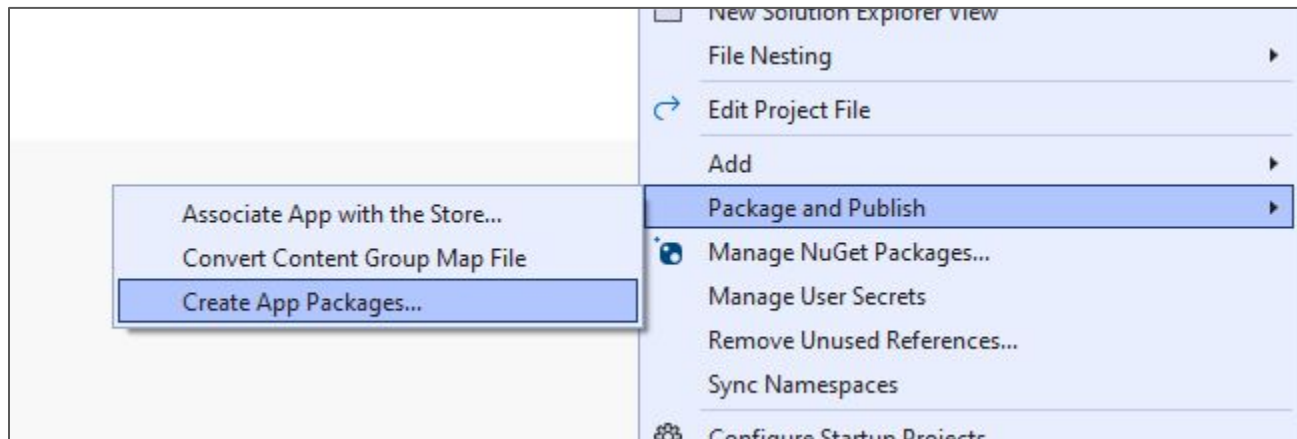
Select distribution method

How will you distribute this application?

○ Microsoft Store under a new app name

◉ Sideloading What is sideloading?

☑ Enable automatic updates

Sign your package & Trust it

Select signing method

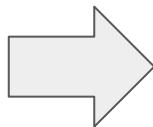Would you like to sign this app package?

○ No, skip package signing.
App packages must be signed before they can be installed. You will have to sign the app packages externally
How do I sign an app package?

◉ Yes, select a certificate:

| Select From Azure Key Vault... | Select From Store... | Select From File... | Create... |

❌ Choose a certificate to sign the app package.

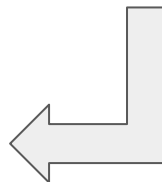## Create a Self-Signed Test Certificate

Self-signed certificates are intended for testing and may prevent your app from installing on other machines. We recommend using a trusted certificate when publishing your app.

Publisher Common Name

| tdqua |

Enter password for new file App1_TemporaryKey.pfx

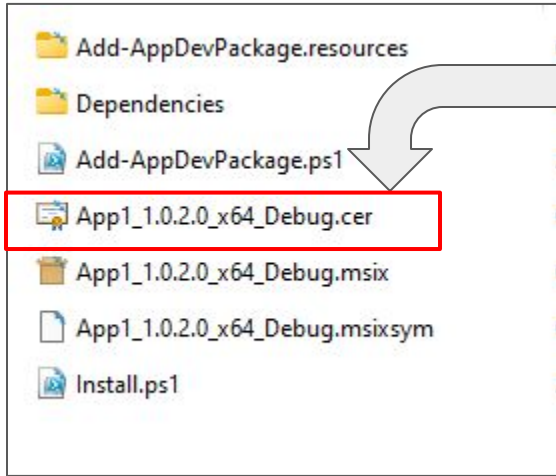Confirm your password

Would you like to sign this app package?

○ No, skip package signing.
App packages must be signed before they can be installed. You will have to sign the app packages externally.
How do I sign an app package?

◉ Yes, use the current certificate:

Subject: CN=tdqua
Thumbprint: 5C7B39DC341DF8F0C60789299031C8D770BE906F
Expiration Date: 04/08/2025 2:50:01 AM

ⓘ This certificate is not trusted on this PC.
If you plan to install this app, import this certificate to the trusted certificate store.
This operation requires Administrator privileges.

Learn about signing certificates.

| Remove | Details |

🛡 Trust

Install using the newly created msix file

Add-AppDevPackage.resources

Dependencies

Add-AppDevPackage.ps1

App1_1.0.2.0_x64_Debug.cer

App1_1.0.2.0_x64_Debug.msix

App1_1.0.2.0_x64_Debug.msixsym

Install.ps1

On other machine, install certificate first!!!! (file .cer)

Template Studio - Fixing RuntimeIdentifier error

The specified RuntimeIdentifier 'win10-arm64' is not recognized. See https://aka.ms/netsdk1083 for more information.

Known RIDs

The following list shows a small subset of the most common RIDs used for each OS. For the latest and complete version, see the PortableRuntimeIdentifierGraph.json in the dotnet/runtime repository.

Windows RIDs

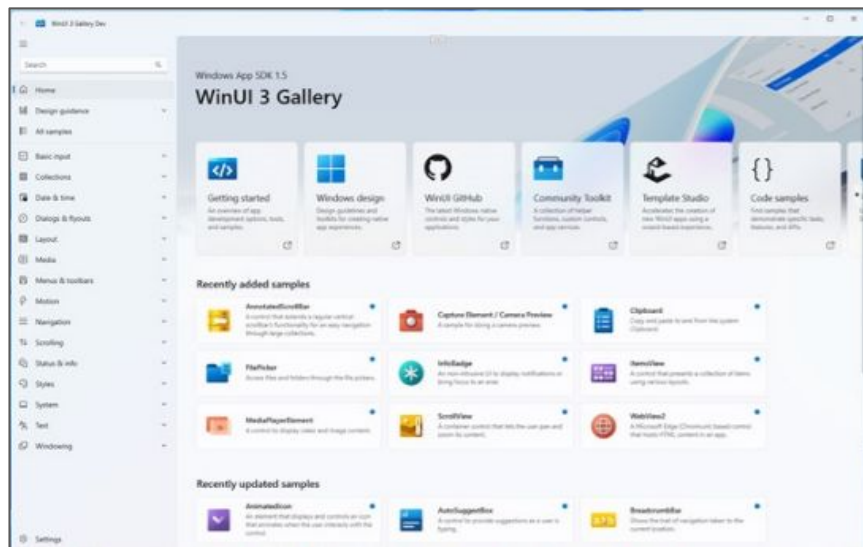- `win-x64`
- `win-x86`
- `win-arm64`

Right click Project > Edit project file, delete the line that contains "win10-x86;win10-x64;win10-arm64"
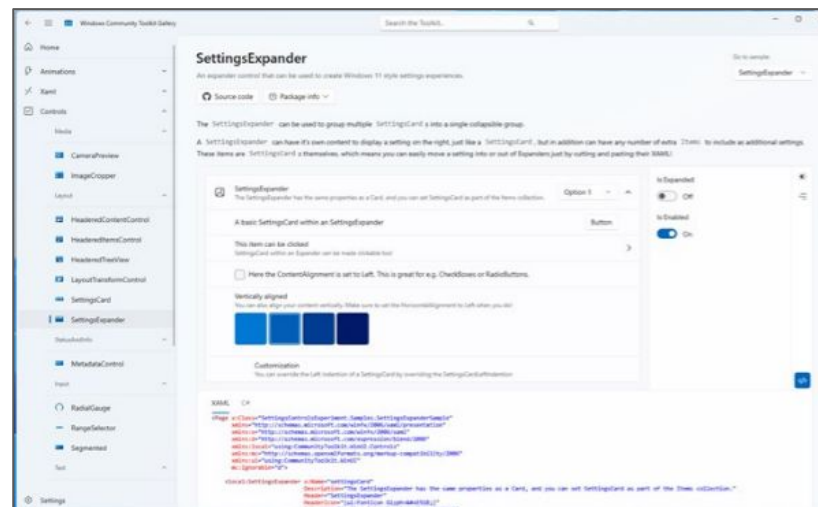
Samples & Resources
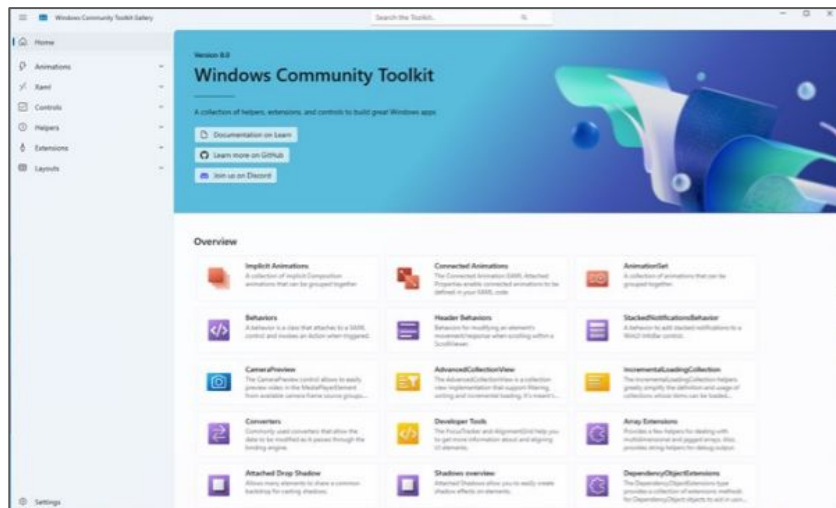
Store > WinUI 3 gallery

❑ Get an idea of what WinUI has to offer

Source: https://github.com/microsoft/WinUI-Gallery

Store > Windows Community Toolkit Gallery

❏ Collection of helper functions, custom controls, app services
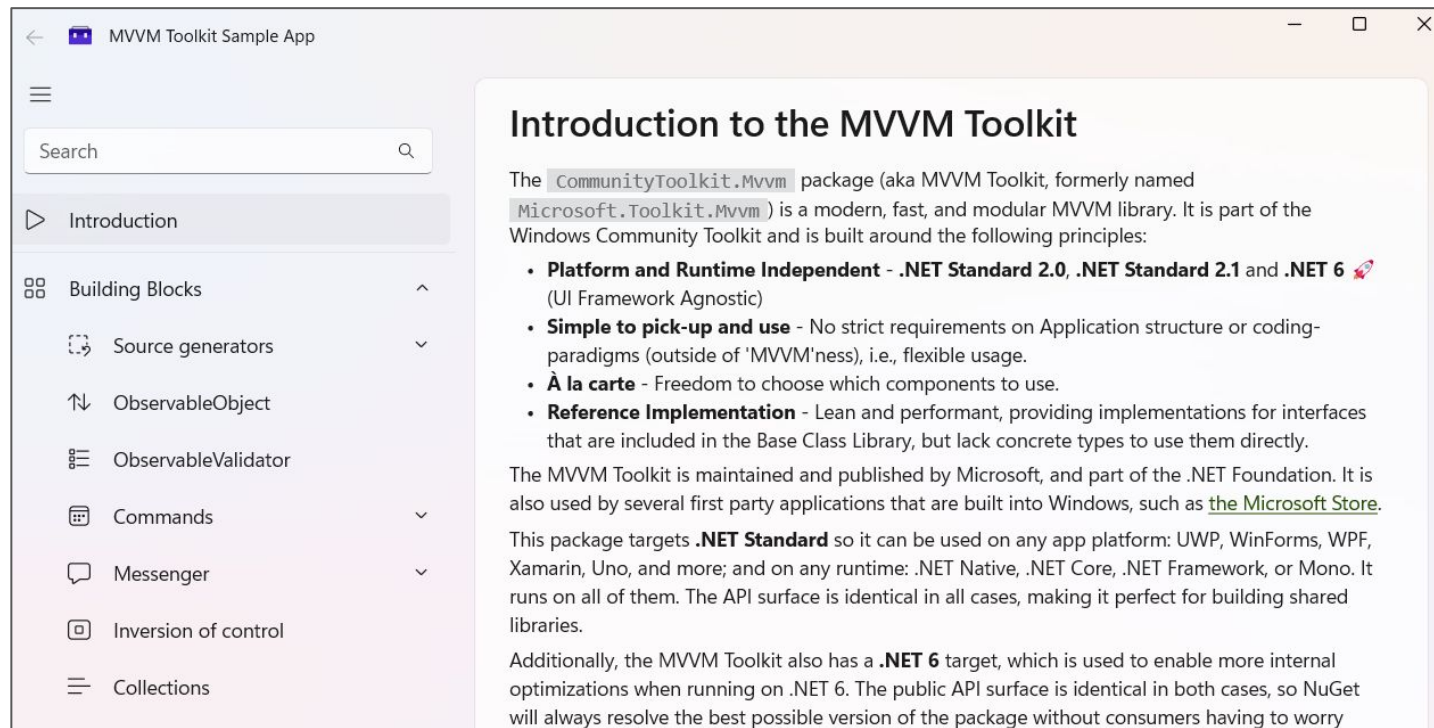❏ Simplifies and demonstrates common developer tasks

Source: https://github.com/CommunityToolkit/Windows

Store > Windows Composition Samples

❏   How to customize your app

Source: https://github.com/microsoft/WindowsCompositionSamples
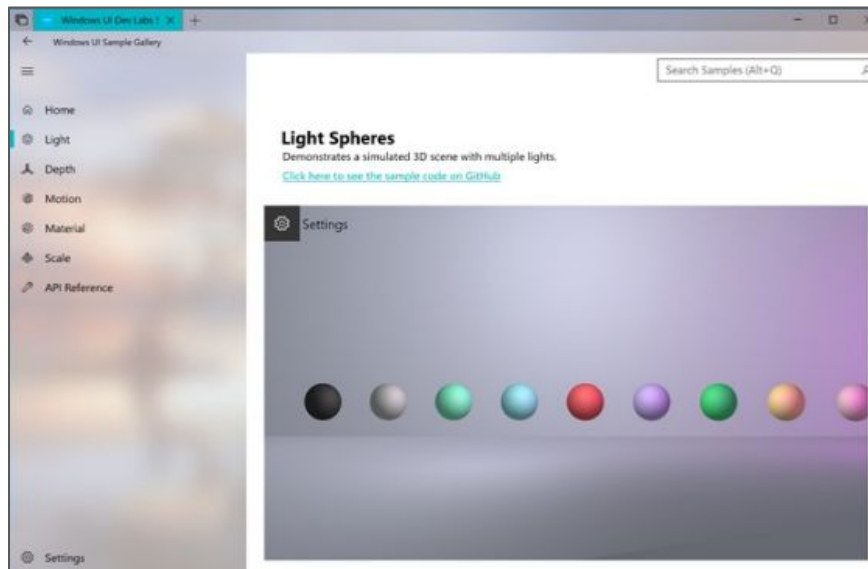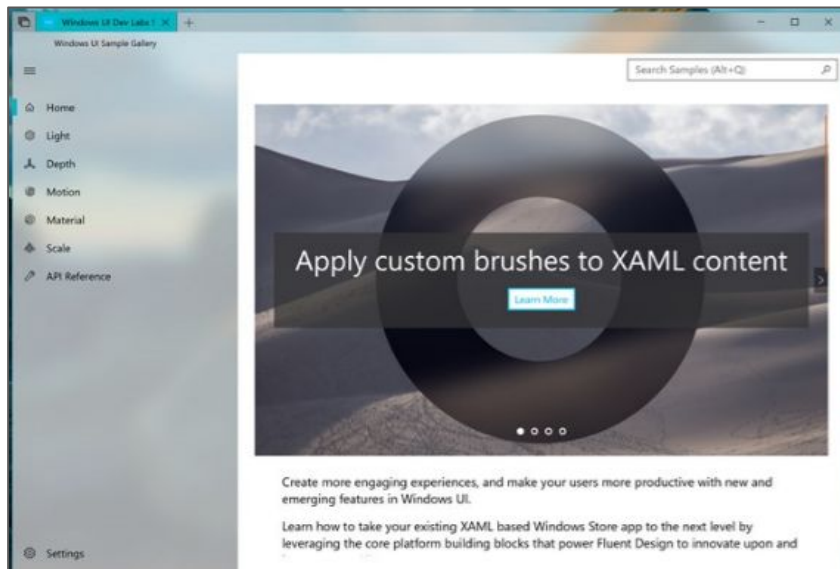
MVVM Toolkit & Samples

[CommunityToolkit/MVVM-Samples: Sample repo for MVVM package](#)

26

More on MVVM here

https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm

Smallest MVVM example:

https://github.com/MarkWithall/worlds-simplest-csharp-wpf-mvvm-example/tree/C%239.0

MVC or MVVM?
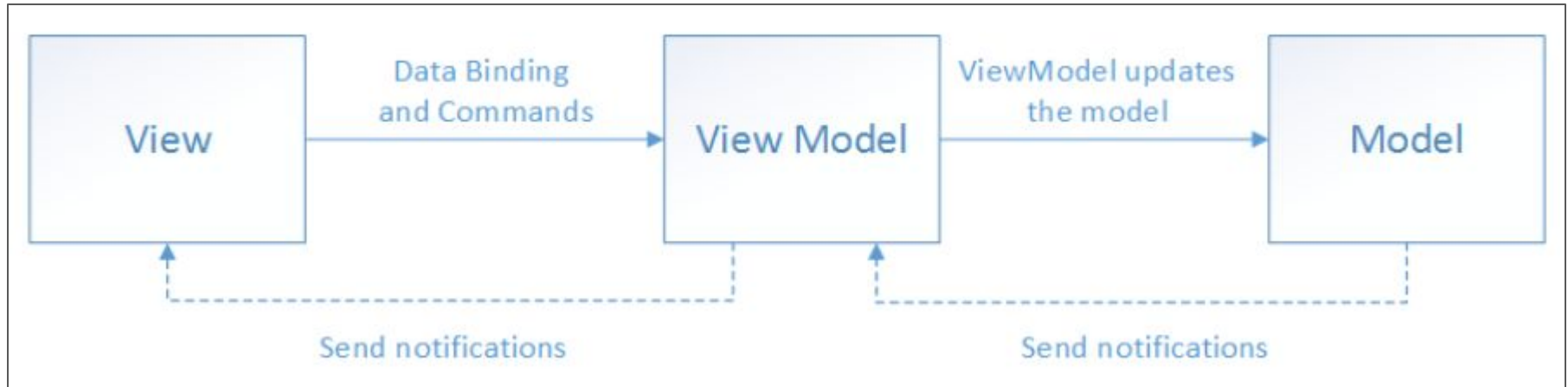
https://stackoverflow.com/questions/667781/what-is-the-difference-between-mvc-and-mvvm

More samples source code

- https://github.com/microsoft/WindowsAppSDK-Samples

- https://github.com/scottkuhl/awesome-winui

- https://learn.microsoft.com/en-us/windows/apps/design/downloads

Best practices

User experience

❏ [Windows 11 design principles](#)
   ❏ [WinUI 3 Figma](#) Design toolkit

Areas to focus
1. Layout
2. UI interaction
3. Visual style
4. Window behavior
5. Shell integration points

Layout

❏　Test your application: dimensions, devices, window sizes, DPI settings, and scale settings

❏　Should work as expected even when resized down to small dimensions

Responsive layout

https://learn.microsoft.com/en-us/windows/apps/design/layout/responsive-design

| Size class	Breakpoints	Typical screen size	Devices	Window Sizes
Small	up to 640px	20" to 65"	TVs	320x569, 360x640, 480x854
Medium	641 - 1007px	7" to 12"	Tablets	960x540
Large	1008px and up	13" and up	PCs, Laptops, Surface Hub	1024x640, 1366x768, 1920x1080

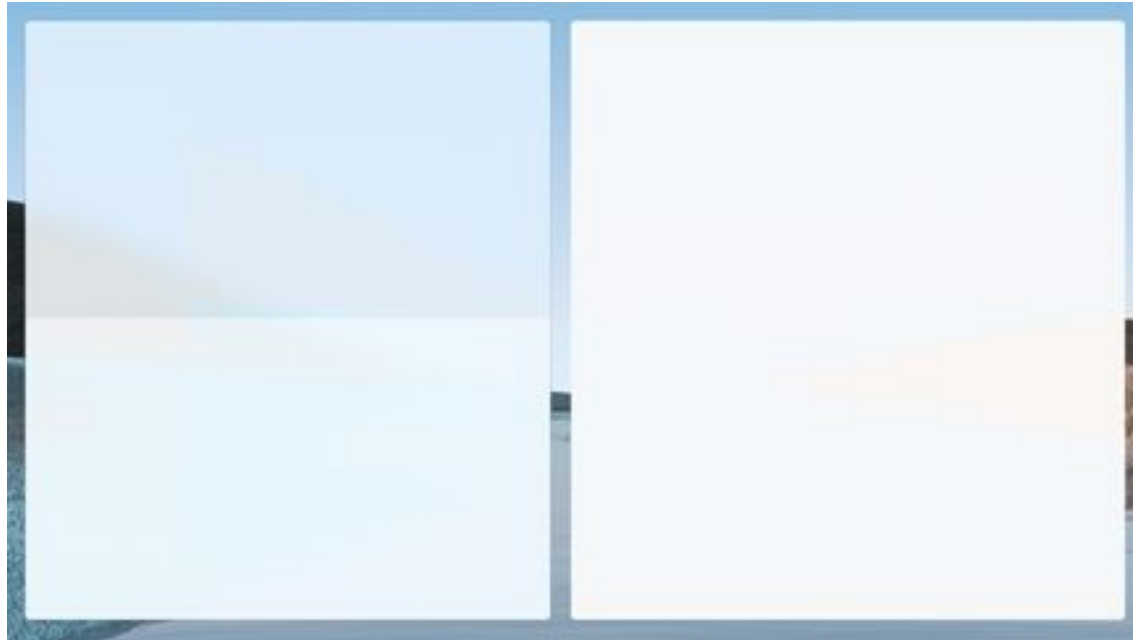Reposition

Resize

Reflow

Show/hide

Replace

Re-architect

On-object commanding
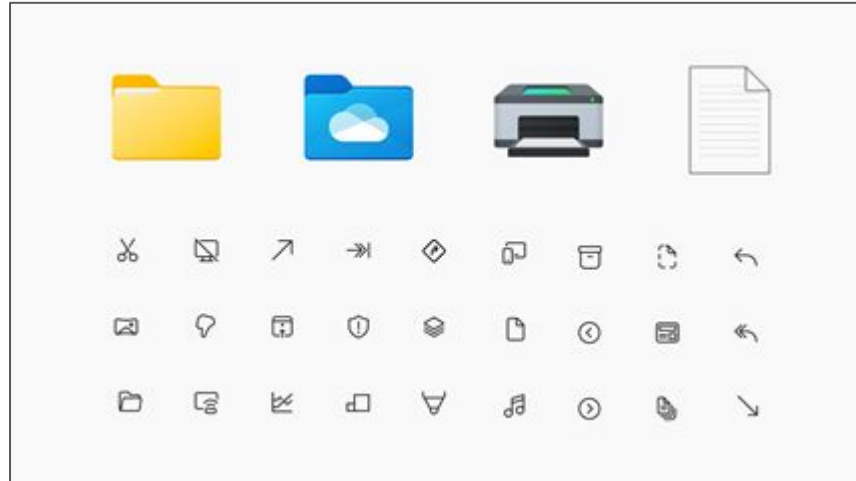
1. Context menus
2. Swipe commands
3. Keyboard shortcuts.
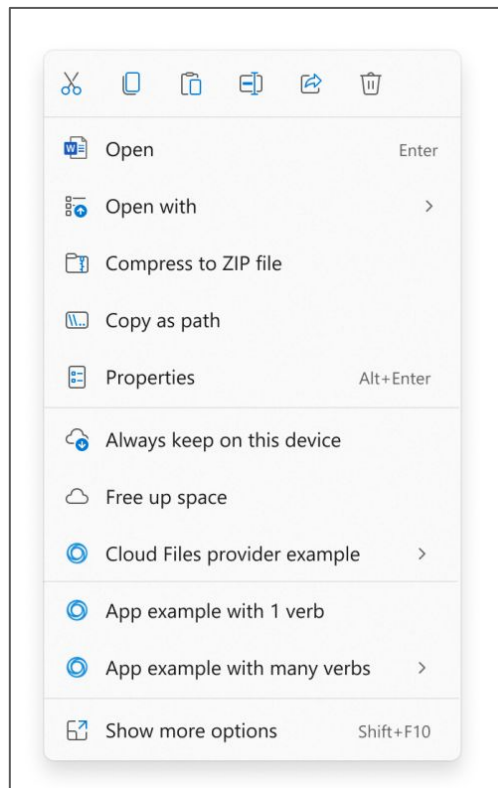
Materials: Acrylic and Mica

[11 things to make your app great](#)

Iconography and typography

New fonts & New icons

Clone war

Apps to clone as a practice

❏ Store

❏ Settings

❏ Weather

❏ XBox / Netflix / Spotify