Windows App SDK 1.5

WinUI 3 Gallery



Getting started

An overview of app development options, tools, and samples.





Windows design

Design guidelines and toolkits for creating native app experiences.





WinUI GitHub

The latest Windows na 5. controls and styles for applications.

- 1. WinUI 3 简介
- 项目模板
- 3. XAML 基础语法
 - 3.1 常用 XAML 控件
 - 3.2 数据绑定
- 事件处理
- 控件属性
- 布局系统
 - 6.1 StackPanel 示例
 - 6.2 Grid 示例
- 样式与资源
- 模板与数据模板
 - 8.1 Button 控件模板

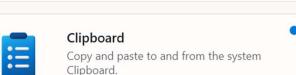
- 9. **命令与 MVVM** 模式
 - 9.1 ICommand 实现
- 10. 导航系统
 - 10.1 Frame 示例
- WinRT API
- 11.1 文件访问示例
- 12. **自定义控件**
 - 12.1 自定义控件示例
- 13. 本地化与多语言支持
- 14. 调试与部署
- 15. 性能优化

Recently added samples



AnnotatedScrollBar

A control that extends a regular vertical scrollbar's functionality for an easy navigation through large collections.





Capture Eleme 7.



1. WinUI 3 简介

- WinUI 3 是微软为现代 Windows 应用提供的用户界面框架,基于 XAML 和 C++/C#。
- 用于创建桌面应用 (Win32 应用) 或 UWP 应用。

2. 项目模板

- Visual Studio 支持 WinUI 3 项目模板:
 - C++/WinRT 项目。
 - C# 项目。

3. XAML 基础语法

• WinUI 3 使用 XAML 来定义 UI。

3.1 常用 XAML 控件:

Button:

```
xml

Copy code

<Button Content="Click Me" Click="OnClick"/>
```

TextBlock:

```
xml

Copy code

<TextBlock Text="Hello, WinUI 3!" />
```

• Grid 布局:

3.2 数据绑定

• XAML 支持数据绑定,可以使用 {Binding} 来将数据源绑定到 UI 元素。

```
xml

Copy code

<TextBlock Text="{Binding Name}" />
```

4. 事件处理

• 事件处理器在 XAML 中指定, 在 C++ 或 C# 中实现。

C++ 事件处理器:

```
cpp

void MainWindow::OnButtonClick(IInspectable const&, RoutedEventArgs const&)
{
    // 处理点击事件
}
```

5. 控件属性

• XAML 控件有大量属性可以设置,例如 Width, Height, Margin, Padding:

```
xml

<Button Content="Click Me" Width="200" Height="50"/>
```

6. 布局系统

• WinUI 3 提供多种布局容器:

• StackPanel: 垂直或水平堆叠。

• Grid: 基于行列的网格布局。

• RelativePanel: 相对布局。

6.1 StackPanel 示例:

6.2 Grid 示例:

7. 样式与资源

• WinUI 3 支持样式与资源的重用:

8. 模板与数据模板

• 控件可以自定义 ControlTemplate 和 DataTemplate。

8.1 Button 控件模板:

9. 命令与 MVVM 模式

• WinUI 3 支持 MVVM (Model-View-ViewModel) 架构,通过命令和数据绑定实现分离。

9.1 ICommand 实现:

```
Copy code
csharp
public ICommand MyCommand { get; }
public MyViewModel()
   MyCommand = new RelayCommand(ExecuteCommand);
private void ExecuteCommand()
   // 命令执行逻辑
```

10. 导航系统

• WinUI 3 提供 Frame 控件用于页面导航。

10.1 Frame 示例:

```
xml

<Frame x:Name="MainFrame"/>
```

在代码中导航:

```
cpp

MainFrame().Navigate(xaml_typename<MyApp::SecondPage>());
```

11. WinRT API

• WinUI 3 可以与 Windows Runtime API 进行交互,使用 Windows:: 命名空间中的 API。

11.1 文件访问示例:

```
cpp

StorageFolder folder = KnownFolders::DocumentsLibrary();
StorageFile file = co_await folder.CreateFileAsync(L"sample.txt");
```

12. 自定义控件

• 可以通过继承基础控件创建自定义控件。

12.1 自定义控件示例:

```
cpp

public ref class MyCustomButton: public Button
{
    // 自定义逻辑
};
```

13. 本地化与多语言支持

• 支持通过 .resw 文件进行资源本地化。



14. 调试与部署

• 通过 Visual Studio 调试 WinUI 3 应用, 部署为桌面应用或 UWP 应用。

15. 性能优化

• 使用虚拟化控件如 VirtualizingStackPanel 来提高性能。