# Visual Studio 2022扩展开发入门

## 课前准备（1）

1. 认识VS扩展

|  |  |
| --- | --- |
| <Grid>  <Button  Width="85"  **Background="Azure"**  **Foreground="Black"**  Height="42"  Margin="95,102,0,0"  HorizontalAlignment="Left"  VerticalAlignment="Top"  Content="Button" />  </Grid> | <Grid>  <Button  Width="85"  Height="42"  Margin="95,102,0,0"  HorizontalAlignment="Left"  VerticalAlignment="Top"  **Background="Azure"**  Content="Button"  **Foreground="Black"** />  </Grid> |

<https://github.com/Xavalon/XamlStyler/>

1. 扩展开发环境搭建

C:\Users\Zmrbak\.nuget\packages\microsoft.vssdk.buildtools\15.9.3032\tools\VSSDK\Microsoft.VsSDK.targets(752,5): error : Extension '0c572937-0d4a-48cc-aa3f-875ffa9805ba' is of the same or lower version than the installed extension. Please make sure that the extension version is higher than that of the installed extension.

========== 全部重新生成: 成功 2 个，失败 1 个，跳过 0 个 ==========

* 卸载已安装的扩展
* **提供新版本号**

**source.extension.vsixmanifest**

<Identity Id="**8C9E25DA-8380-484D-8BDC-5D8120A07E19**" Version="3.2.0.0" Language="en-US" Publisher="Xavalon" />

* 不予理会

1. **Visual Studio 2022** <http://www.microsoft.com>

工作负荷：**Visual Studio 扩展开发**

5>C:\Users\Zmrbak\.nuget\packages\microsoft.vssdk.buildtools\15.9.3032\tools\VSSDK\Microsoft.VsSDK.targets(87,5): error MSB4062: 未能从程序集 C:\Users\Zmrbak\.nuget\packages\microsoft.vssdk.buildtools\15.9.3032\tools\VSSDK\Microsoft.VisualStudio.Sdk.BuildTasks.15.0.dll 加载任务“CompareBuildTaskVersion”。

Could not load file or assembly 'file:///C:\Users\Zmrbak\.nuget\packages\microsoft.vssdk.buildtools\15.9.3032\tools\VSSDK\Microsoft.VisualStudio.Sdk.BuildTasks.15.0.dll' or one of its dependencies.

试图加载格式不正确的程序。 请确认 <UsingTask> 声明正确，该程序集及其所有依赖项都可用，并且该任务包含实现 Microsoft.Build.Framework.ITask 的公共类。

1. 本课程目的：移植XamlStyler扩展到Visual Studio 2022
2. 课程源代码：<https://github.com/zmrbak/VSIX2022>
3. 如何下载课程源代码

## 课前准备（2）

## VSIX项目文件结构（1）

目标框架：.NET Framework 4.7.2

<Assets>

<Asset

Type="**Microsoft.VisualStudio.VsPackage**"

d:Source="Project"

d:ProjectName="%CurrentProject%"

Path="|%CurrentProject%;PkgdefProjectOutputGroup|" />

</Assets>

C:\Users\lib\**.nuget**\packages\microsoft.visualstudio.shell.15.0\17.0.0-previews-1-31410-258\lib\net472\Microsoft.VisualStudio.**Shell**.15.0.dll

## VSIX项目文件结构（2）

## 向VS菜单中添加菜单项（1）

Microsoft.VisualStudio.Interop.dll

namespace Microsoft.VisualStudio.Shell.Interop

{

public enum OLEMSG**ICON**

{

OLEMSGICON\_NOICON,

OLEMSGICON\_CRITICAL,

OLEMSGICON\_QUERY,

OLEMSGICON\_WARNING,

OLEMSGICON\_INFO

}

}

namespace Microsoft.VisualStudio.Shell.Interop

{

public enum OLEMSG**BUTTON**

{

OLEMSGBUTTON\_OK,

OLEMSGBUTTON\_OKCANCEL,

OLEMSGBUTTON\_ABORTRETRYIGNORE,

OLEMSGBUTTON\_YESNOCANCEL,

OLEMSGBUTTON\_YESNO,

OLEMSGBUTTON\_RETRYCANCEL,

OLEMSGBUTTON\_YES**ALL**NOCANCEL

}

}

namespace Microsoft.VisualStudio.Shell.Interop

{

public enum OLEMSG**DEF**BUTTON

{

OLEMSGDEFBUTTON\_FIRST,

OLEMSGDEFBUTTON\_SECOND,

OLEMSGDEFBUTTON\_THIRD,

OLEMSGDEFBUTTON\_FOURTH

}

}

C:\Program Files\Microsoft Visual Studio\2022\Preview\**VS**SDK\VisualStudioIntegration\Common\Inc

## 向VS菜单中添加菜单项（2）

## 向VS菜单中添加菜单项（3）

## 向VS中注册一个事件（1）

Comment Remover

<https://github.com/madskristensen/CommentRemover>

[ProvideAutoLoad(VSIX08Package.UIContextGuidString, PackageAutoLoadFlags.BackgroundLoad)]

[ProvideUIContextRule(VSIX08Package.UIContextGuidString, name: "XAML load", expression: "Dotxaml", termNames: new[] { "Dotxaml" }, termValues: new[] { "HierSingleSelectionName:.xaml$" })]

VSConstants

CommandEvents

DTE

## 向VS中注册一个事件（2）

## 命令菜单组与命令ID

<https://github.com/madskristensen/ExtensibilityTools>

HKEY\_CURRENT\_USER\SOFTWARE\Microsoft\VisualStudio\17.0abc\General

EnableVSIPLogging DWORD with value 1

关闭注册表编辑器

重新启动VS2022

Hold Ctrl + Shift and interact with IDE

<https://github.com/madskristensen/CommandTableInfo/>

<GuidSymbol name="guidVSIX10PackageCmdSet" value="{5f0f5edc-575e-4d1e-8696-af66dc792868}">

<IDSymbol name="MyMenuGroup" value="0x1020" />

<IDSymbol name="Command1Id" value="0x0100" />

</GuidSymbol>

public static readonly Guid CommandSet = new Guid("5f0f5edc-575e-4d1e-8696-af66dc792868");

5f0f5edc-575e-4d1e-8696-af66dc792868

5f0f5edc-575e-4d1e-8696-af66dc792868

public const int CommandId = 0x0100;

<IDSymbol name="Command1Id" value="0x0100" />

## VS扩展开发“扩展套装”：Extensibility Essentials（1）

Extensibility Essentials 2022

1. VSIX Synchronizer
2. Insert Guid
3. Image Optimizer
4. Command Explorer
5. Settings Store Explorer
6. KnownMonikers Explorer

**<Include href="KnownImageIds.vsct"/>**

<Button guid="guidVSIX11PackageCmdSet" id="Command1Id" priority="0x0100" type="Button">

<Parent guid="guidVSIX11PackageCmdSet" id="MyMenuGroup" />

**<Icon guid="ImageCatalogGuid" id="Action" />**

**<CommandFlag>IconIsMoniker</CommandFlag>**

<Strings>

<ButtonText>VSXI11.1 Command</ButtonText>

</Strings>

</Button>

1. Clean MEF Component Cache
2. VSCT IntelliSense
3. Image Manifest Tools
4. **Extensibility Logs**
5. Pkgdef Language
6. Extensibility Template Pack

Comment Remover

## VS扩展开发“扩展套装”：Extensibility Essentials（2）

## VS扩展开发“扩展套装”：Extensibility Essentials（3）

## 向菜单命令绑定键盘快捷键

CTRL+1,CTRL+2

<KeyBindings>

<KeyBinding guid="guidVSIX14Package**CmdSet**" id="**Command1Id**" editor="**guidVSStd97**" key1="1" mod1="CONTROL" key2="2" mod2="CONTROL">

</KeyBinding>

</KeyBindings>

## 向VS菜单添加多条命令(1)

%comspec% /C "C:\Program Files\Microsoft Visual Studio\2022\Preview\VSSDK\VisualStudioIntegration\Tools\Bin\Create**Exp**Instance.exe" /Reset /VSInstance=17.0\_dee98566 /RootSuffix=Exp && PAUSE

* Removed directory C:\Users\lib\AppData\Local\Microsoft\VisualStudio\17.0\_dee98566Exp.
* Created directory C:\Users\lib\AppData\Local\Microsoft\VisualStudio\17.0\_dee98566Exp\Extensions.
* Copied Extensions directory from C:\Users\lib\AppData\Local\Microsoft\VisualStudio\17.0\_dee98566\Extensions to C:\Users\lib\AppData\Local\Microsoft\VisualStudio\17.0\_dee98566Exp\Extensions\Extensions-17.0\_dee98566.

**VSIX15Package.vsct**

<Symbols>

<GuidSymbol name="guidVSIX15PackageCmdSet" value="{9a9f54d2-caca-4df5-b9c2-afedd6b99890}">

**<IDSymbol name="cmdidCommand3" value="0x1022"/>**

</GuidSymbol>

</Symbols>

<Buttons>

<Button guid="guidVSIX15PackageCmdSet" id="**cmdidCommand3**" priority="0x0100" type="Button">

<Parent guid="guidVSIX15PackageCmdSet" id="**MyMenuGroup**" />

<Strings>

<ButtonText>Invoke Command3</ButtonText>

</Strings>

</Button>

</Buttons>

<Groups>

<Group guid="guidVSIX15PackageCmdSet" id="**MyMenuGroup**" priority="0x0600">

<Parent guid="guidSHLMainMenu" id="IDM\_VS\_MENU\_TOOLS" />

</Group>

</Groups>

Group(id) -> Button(id)

**IDSymbol -> id**

VSIX15Package.cs

protected override async Task InitializeAsync(CancellationToken cancellationToken, IProgress<ServiceProgressData> progress)

{

await this.JoinableTaskFactory.SwitchToMainThreadAsync(cancellationToken);

await Command1.InitializeAsync(this);

await Command2.InitializeAsync(this);

**await Command3.InitializeAsync(this);**

}

Command3.cs

internal sealed class Command3｛

｝

Priority 决定命令的排列顺序，数字越小，则越靠上，数字越大越靠下。

## 向VS菜单添加多条命令(2)

## 向VS菜单添加多级菜单

Group

Button

Button

Group

Menu

Group

Group

Button

## 命令组与命令的重复使用

<CommandPlacements>

<CommandPlacement guid="guidVSIX18PackageCmdSet" id="cmdidCommand3" priority="100">

<Parent guid="guidVSIX18PackageCmdSet" id="**MyMenuGroup1**" />

</CommandPlacement>

</CommandPlacements>

## 向子菜单添加动态菜单（1）

<Button guid="guidVSIX19PackageCmdSet" id="Command1Id" priority="0x0100" type="Button">

<Parent guid="guidVSIX19PackageCmdSet" id="MRUGroup" />

**<CommandFlag>DynamicItemStart</CommandFlag>**

<Strings>

**<ButtonText></ButtonText>**

</Strings>

</Button>

//动态菜单数量

private int mruNumber = 10;

//菜单名称列表

private List<string> mruList;

private void InitMruMenu(OleMenuCommandService commandService)

{

mruList = new List<string>();

for (int i = 0; i < mruNumber; i++)

{

string message = string.Format(CultureInfo.CurrentCulture, "Item {0}", i + 1);

mruList.Add(message);

var menuCommandID = new CommandID(CommandSet, CommandId + i);

var menuItem = new **OleMenuCommand**(this.Execute, menuCommandID);

menuItem.Text = message;

menuItem.**BeforeQueryStatus** += MenuItem\_BeforeQueryStatus;

commandService.AddCommand(menuItem);

}

}

## 向子菜单添加动态菜单（2）

## 更改VS扩展的菜单标题和状态（1）

<Button guid="guidVSIX21PackageCmdSet" id="Command1Id" priority="0x0100" type="Button">

<Parent guid="guidVSIX21PackageCmdSet" id="MyMenuGroup" />

**<CommandFlag>TextChanges</CommandFlag>**

<Strings>

<ButtonText>VSIX21 Invoke Command1</ButtonText>

</Strings>

</Button>

var menuCommandID = new CommandID(CommandSet, CommandId);

var menuItem = new **OleMenuCommand**(this.Execute, menuCommandID);

menuItem.BeforeQueryStatus += MenuItem\_BeforeQueryStatus;

commandService.AddCommand(menuItem);

OleMenuCommand menuCommand = **sender as OleMenuCommand**;

menuCommand.Text = "item" + index;

var menuCommandID = new CommandID(CommandSet, PackageIds.cmdidCommand2);

var commandService = package.**GetService**<IMenuCommandService, OleMenuCommandService>();

var menuCommand1 = commandService.FindCommand(menuCommandID) as **OleMenuCommand**;

menuCommand1.Text = "item" + index;

## 更改VS扩展的菜单标题和状态（1）

## 菜单命令的可见性约束

<VisibilityConstraints>

<VisibilityItem guid="guidVSIX23PackageCmdSet" id="**Command1Id**" context="UICONTEXT\_**SolutionExists**">

</VisibilityItem>

</VisibilityConstraints>

<Button guid="guidVSIX23PackageCmdSet" id="Command1Id" priority="0x0100" type="Button">

<Parent guid="guidVSIX23PackageCmdSet" id="MyMenuGroup" />

<Icon guid="guidImages" id="bmpPic1" />

<CommandFlag>**DynamicVisibility**</CommandFlag>

<Strings>

<ButtonText>VSIX23 Invoke Command1</ButtonText>

</Strings>

</Button>

<Group guid="guidVSIX23PackageCmdSet" id="MyMenuGroup" priority="0x0600">

<Parent guid="guidSHLMainMenu" id="**IDM\_VS\_CTXT\_ITEMNODE**"/>

</Group>

var menuCommandID = new CommandID(CommandSet, CommandId);

var menuItem = new **OleMenuCommand**(this.Execute, menuCommandID) { **Supported** = false };

[ProvideUIContextRule(

contextGuid: VSConstants.UICONTEXT.**SolutionExists\_string**,

name: "CSharp",

expression: "VB | CS",

termNames: new string[] { "VB", "CS" },

termValues: new string[] { "HierSingleSelectionName:.vb$","HierSingleSelectionName:.cs$" }

)]

## 向VS工具栏添加按钮

<Group guid="guidVSIX24PackageCmdSet" id="MyMenuGroup" priority="0x0600">

<Parent guid="**VSMainMenu**" id="**StandardToolbar**" />

</Group>

<Menu guid="guidVSIX24PackageCmdSet" id="**MyMenu**" priority="100" type="**MenuController**">

<Parent guid="guidVSIX24PackageCmdSet" id="**MyMenuGroup**"/>

<Strings>

<ButtonText>string</ButtonText>

</Strings>

</Menu>

<Group guid="guidVSIX24PackageCmdSet" id="**SubMenuGroup**" priority="100">

<Parent guid="guidVSIX24PackageCmdSet" id="**MyMenu**"/>

</Group>

<Button guid="guidVSIX24PackageCmdSet" id="cmdidCommand2" priority="0x0100" type="Button">

<Parent guid="guidVSIX24PackageCmdSet" id="**SubMenuGroup**" />

<Icon guid="guidImages1" id="bmpPic1" />

<CommandFlag>**FixMenuController**</CommandFlag>

<Strings>

<ButtonText>Invoke Command2</ButtonText>

</Strings>

</Button>

## 向VS添加可停靠工具栏

<Menus>

<Menu guid="guidVSIX25PackageCmdSet" id="MyMenu" priority="100" type="**Toolbar**">

<CommandFlag>**DefaultDocked**</CommandFlag>

<Strings>

<ButtonText>测试 ToolBar</ButtonText>

</Strings>

</Menu>

</Menus>

<IDSymbol name="MyMenu" value="0x0101" />

<Group guid="guidVSIX25PackageCmdSet" id="**MyMenuGroup**" priority="0x0600">

<Parent guid="guidVSIX25PackageCmdSet" id="**MyMenu**"/>

</Group>

<Button guid="guidVSIX25PackageCmdSet" id="Command1Id" priority="0x0100" type="Button">

<Parent guid="guidVSIX25PackageCmdSet" id="**MyMenuGroup**" />

<Icon guid="guidImages" id="bmpPic1" />

<Strings>

<ButtonText>Invoke 测试 Command1Id</ButtonText>

</Strings>

</Button>

## 向VS添加工具窗口（1）

<https://www.scootersoftware.com/>

//异步

private void **Execute**(object sender, EventArgs e)

{

this.package.JoinableTaskFactory.**RunAsync**(async delegate

{

ToolWindowPane window = await this.package.ShowToolWindowAsync(typeof(ToolWindow1), 0, true, this.package.DisposalToken);

if ((null == window) || (null == window.Frame))

{

throw new NotSupportedException("Cannot create tool window");

}

});

}

//同步

private void **Execute**(object sender, EventArgs e)

{

ThreadHelper.ThrowIfNotOnUIThread();

ToolWindowPane window = this.package.**FindToolWindow**(typeof(ToolWindow1), 0, true);

if ((null == window) || (null == window.Frame))

{

throw new NotSupportedException("Cannot create tool window");

}

IVsWindowFrame windowFrame = (IVsWindowFrame)window.Frame;

Microsoft.VisualStudio.ErrorHandler.ThrowOnFailure(windowFrame.Show());

}

<Button guid="guidVSIX26PackageCmdSet" id="ToolWindow1CommandId" priority="0x0100" type="Button">

<Parent guid="guidSHLMainMenu" id="**IDG\_VS\_WNDO\_OTRWNDWS1**"/>

<Icon guid="guidImages" id="bmpPic1" />

<Strings>

<ButtonText>VSIX26 ToolWindow1</ButtonText>

</Strings>

</Button>

Button-> Execute-> ToolWindow1-> ToolWindow1Control->XAML/CS

## 向VS添加工具窗口（2）

## 为新添加的工具窗口添加工具栏（1）

<Menus>

<Menu guid="guidVSIX28PackageCmdSet" id="**MyMenu**" priority="100" type="**ToolWindowToolbar**">

<CommandFlag>**DefaultDocked**</CommandFlag>

<Strings>

<ButtonText>测试 Toolbar</ButtonText>

<CommandName>测试 Toolbar</CommandName>

</Strings>

</Menu>

</Menus>

<Groups>

<Group guid="guidVSIX28PackageCmdSet" id="MyMenuGroup" priority="0x0600">

<Parent guid="guidVSIX28PackageCmdSet" id="**MyMenu**" />

</Group>

</Groups>

<Button guid="guidVSIX28PackageCmdSet" id="Command1Id" priority="0x0100" type="Button">

<Parent guid="guidVSIX28PackageCmdSet" id="**MyMenuGroup**" />

<Icon guid="guidImages" id="bmpPicX" />

<Strings>

<ButtonText>VSIX28 Invoke Command1</ButtonText>

</Strings>

</Button>

<IDSymbol name="MyMenu" value="0x0101" />

<IDSymbol name="Command1Id2" value="0x0102" />

public ToolWindow1() : base(null)

{

this.Caption = "ToolWindow1";

this.Content = new ToolWindow1Control();

**this.ToolBar = new System.ComponentModel.Design.CommandID(new Guid("4ba44c83-504d-4b54-bfd3-27f478562ab3"), 0x0101);**

}

## 为新添加的工具窗口添加工具栏（2）

## 为新添加的工具窗口添加菜单控制器

## 为VS添加多实例工具窗口

[ProvideToolWindow(typeof(ToolWindow1), **MultiInstances** = true)]

private void Execute(object sender, EventArgs e)

{

**\_ =** this.package.JoinableTaskFactory.RunAsync(async delegate

{

for (int i = 0; **i** < 10; i++)

{

ToolWindowPane window = await this.package.ShowToolWindowAsync(typeof(ToolWindow1), **i**, **false**, this.package.DisposalToken);

if(window==null)

{

window = await this.package.ShowToolWindowAsync(typeof(ToolWindow1), **i**, **true**, this.package.DisposalToken);

if ((null == window) || (null == window.Frame))

{

throw new NotSupportedException("Cannot create tool window");

}

window.Caption = "Window: " + i;

break;

}

}

});

}

## 让VS动态打开添加的工具窗口

[ProvideToolWindowVisibility(typeof(ToolWindow1),VSConstants.UICONTEXT.SolutionExists\_string)]

临时性窗口，该窗关闭后，并重新打开 Visual Studio 时并不会保持。