using System;

using Extensibility;

using EnvDTE;

using EnvDTE80;

using Microsoft.VisualStudio.CommandBars;

using System.Resources;

using System.Reflection;

using System.Globalization;

using System.Windows.Forms;

namespace Mytest1

{

/// 用于实现外接程序的对象。

///

public class Connect : IDTExtensibility2, IDTCommandTarget

{

private DTE2 \_applicationObject;

private AddIn \_addInInstance;

/// 实现外接程序对象的构造函数。请将您的初始化代码置于此方法内。

private InternalInfo internalInfo = InternalInfo.GetInternalInfoInstance();

public Connect()

{

}

/// 实现 IDTExtensibility2 接口的 OnConnection 方法。接收正在加载外接程序的通知。

/// 宿主应用程序的根对象。

/// 描述外接程序的加载方式。

/// 表示此外接程序的对象。

///

public void OnConnection(object application, ext\_ConnectMode connectMode, object addInInst, ref Array custom)

{

\_applicationObject = (DTE2)application;

\_addInInstance = (AddIn)addInInst;

if (connectMode == ext\_ConnectMode.ext\_cm\_UISetup)

{

object[] contextGUIDS = new object[] { };

Commands2 commands = (Commands2)\_applicationObject.Commands;

// 插件命令到快捷菜单

CommandBars commandBars = ((CommandBars)\_applicationObject.CommandBars);

CommandBar commandBar = (CommandBar)commandBars["Code Window"];

this.internalInfo.Pop = (CommandBarPopup)(CommandBarControl)commandBar.Controls.Add( MsoControlType.msoControlPopup, 1, null, 1, true );

this.internalInfo.Pop.Caption = "SearchOnWeb"; //菜单项显示内容

this.internalInfo.Pop.Enabled = true; //可用

this.internalInfo.Pop.Visible = true; //可见

this.internalInfo.Pop.BeginGroup = true;//添加分隔线

try

{

Command searchGoogle = commands.AddNamedCommand2(

\_addInInstance,

"SearchOnGoogle",

"&Google...",

"Search on Google",

true,

0,

ref contextGUIDS,

(int)vsCommandStatus.vsCommandStatusSupported +

(int)vsCommandStatus.vsCommandStatusEnabled,

(int)vsCommandStyle.vsCommandStylePictAndText,

vsCommandControlType.vsCommandControlTypeButton

);

searchGoogle.AddControl(this.internalInfo.Pop.CommandBar, 1);

}

catch (Exception err)

{

MessageBox.Show(err.Message);

}

}

}

/// 实现 IDTExtensibility2 接口的 OnDisconnection 方法。接收正在卸载外接程序的通知。

/// 描述外接程序的卸载方式。

/// 特定于宿主应用程序的参数数组。

///

public void OnDisconnection(ext\_DisconnectMode disconnectMode, ref Array custom)

{

}

/// 实现 IDTExtensibility2 接口的 OnAddInsUpdate 方法。当外接程序集合已发生更改时接收通知。

/// 特定于宿主应用程序的参数数组。

///

public void OnAddInsUpdate(ref Array custom)

{

}

/// 实现 IDTExtensibility2 接口的 OnStartupComplete 方法。接收宿主应用程序已完成加载的通知。

/// 特定于宿主应用程序的参数数组。

///

public void OnStartupComplete(ref Array custom)

{

}

/// 实现 IDTExtensibility2 接口的 OnBeginShutdown 方法。接收正在卸载宿主应用程序的通知。

/// 特定于宿主应用程序的参数数组。

///

public void OnBeginShutdown(ref Array custom)

{

}

/// 实现 IDTCommandTarget 接口的 QueryStatus 方法。此方法在更新该命令的可用性时调用

/// 要确定其状态的命令的名称。

/// 该命令所需的文本。

/// 该命令在用户界面中的状态。

/// neededText 参数所要求的文本。

///

public void QueryStatus(string commandName, vsCommandStatusTextWanted neededText, ref vsCommandStatus status, ref object commandText)

{

if (neededText == vsCommandStatusTextWanted.vsCommandStatusTextWantedNone)

{

if (commandName == "Mytest1.Connect.Mytest1")

{

status = (vsCommandStatus)vsCommandStatus.vsCommandStatusSupported | vsCommandStatus.vsCommandStatusEnabled;

return;

}

}

}

/// 实现 IDTCommandTarget 接口的 Exec 方法。此方法在调用该命令时调用。

/// 要执行的命令的名称。

/// 描述该命令应如何运行。

/// 从调用方传递到命令处理程序的参数。

/// 从命令处理程序传递到调用方的参数。

/// 通知调用方此命令是否已被处理。

///

public void Exec(string commandName, vsCommandExecOption executeOption, ref object varIn, ref object varOut, ref bool handled)

{

handled = false;

if (executeOption == vsCommandExecOption.vsCommandExecOptionDoDefault)

{

if (commandName == "Mytest1.Connect.Mytest1")

{

TextDocument textDoc = (TextDocument)this.\_applicationObject.ActiveDocument.Object("TextDocument");

EditPoint editPoint = textDoc.Selection.ActivePoint.CreateEditPoint(); // 获取当前文档光标所在位置

EditPoint endPoint = textDoc.EndPoint.CreateEditPoint();

Form1 frm = new Form1();

frm.ShowDialog();

editPoint.Insert(PublicVar.sCode); // 在光标位置插入字符串

endPoint.EndOfDocument();

editPoint.StartOfDocument();

editPoint.SmartFormat(endPoint);

handled = true;

return;

}

}

}

//private DTE2 \_applicationObject;

//private AddIn \_addInInstance;

}

}