SW SOLIDWORKS

ATERO Solutions

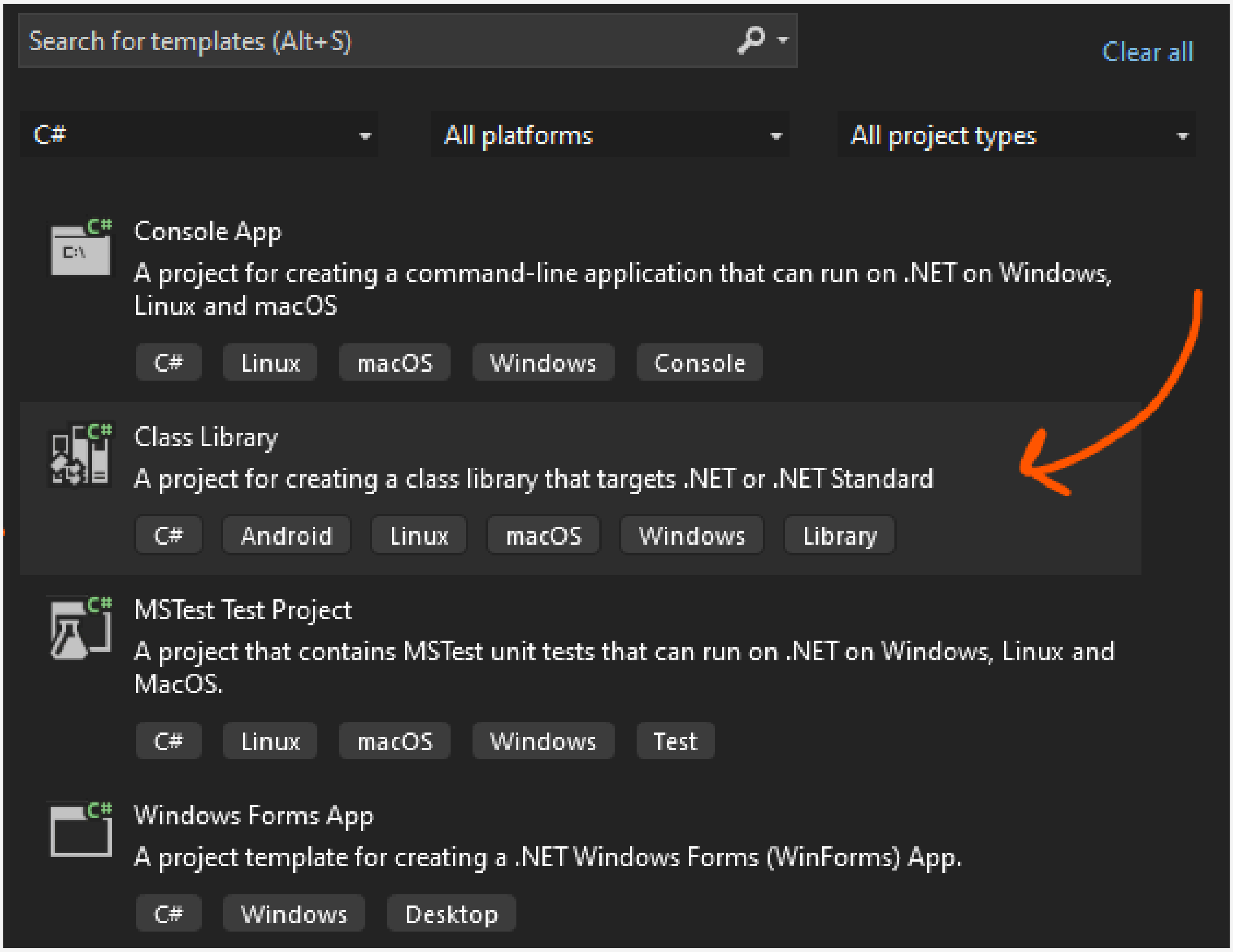
Mastering SOLIDWORKS API

DEPLOY YOUR ADD-IN TO THE TASK PANE

Ahmed Hamed
SOLIDWORKS Expert
info@aterosolutions.com

#01 PREPARE THE PROJECT

- In this tutorial we will use C# language.
- Open anew visual studio project, and make sure you select the “class library”.
- give your project a suitable name. lets assume it will be “test project”.



Search for templates (Alt+S)
Clear all
C#
All platforms
All project types

Console App
A project for creating a command-line application that can run on .NET on Windows, Linux and macOS

C# Linux macOS Windows Console

Class Library
A project for creating a class library that targets .NET or .NET Standard

C# Android Linux macOS Windows Library

MSTest Test Project
A project that contains MSTest unit tests that can run on .NET on Windows, Linux and MacOS.

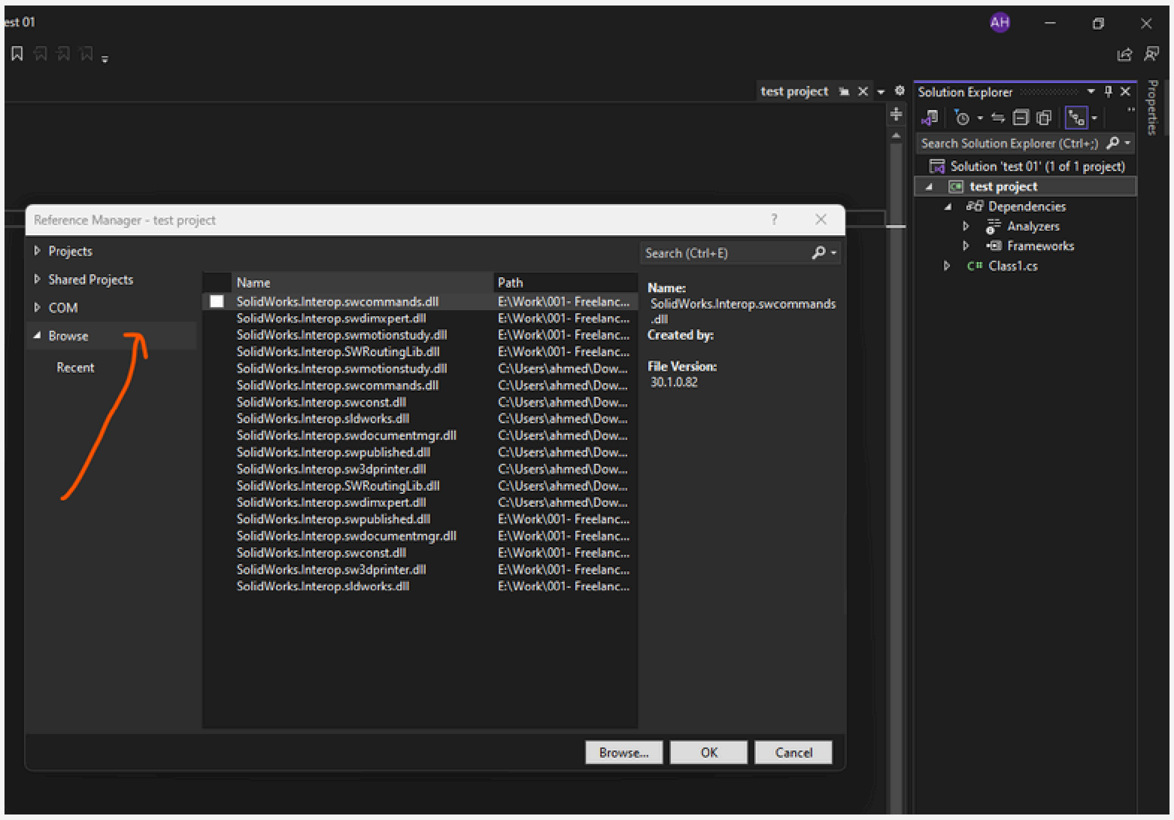
C# Linux macOS Windows Test

Windows Forms App
A project template for creating a .NET Windows Forms (WinForms) App.

Ahmed Hamed
SOLIDWORKS Expert
info@aterosolutions.com

#02 ADD RESOURCES

- After creating the project, in the left properties pan, press right click on the project name and select Add > Project Reference.



	Name	Path
□	SolidWorks.Interop.swcommands.dll	E:\Work\001- Freelanc...
	SolidWorks.Interop.swdimxpert.dll	E:\Work\001- Freelanc...
	SolidWorks.Interop.swmotionstudy.dll	E:\Work\001- Freelanc...
	SolidWorks.Interop.SWRoutingLib.dll	E:\Work\001- Freelanc...
	SolidWorks.Interop.swmotionstudy.dll	C:\Users\ahmed\Dow...
	SolidWorks.Interop.swcommands.dll	C:\Users\ahmed\Dow...
	SolidWorks.Interop.swconst.dll	C:\Users\ahmed\Dow...
	SolidWorks.Interop.sldworks.dll	C:\Users\ahmed\Dow...
	SolidWorks.Interop.swdocumentmgr.dll	C:\Users\ahmed\Dow...
	SolidWorks.Interop.swpublished.dll	C:\Users\ahmed\Dow...
	SolidWorks.Interop.sw3dprinter.dll	C:\Users\ahmed\Dow...
	SolidWorks.Interop.SWRoutingLib.dll	C:\Users\ahmed\Dow...
	SolidWorks.Interop.swdimxpert.dll	C:\Users\ahmed\Dow...
	SolidWorks.Interop.swpublished.dll	E:\Work\001- Freelanc...
	SolidWorks.Interop.swdocumentmgr.dll	E:\Work\001- Freelanc...
	SolidWorks.Interop.swconst.dll	E:\Work\001- Freelanc...
	SolidWorks.Interop.sw3dprinter.dll	E:\Work\001- Freelanc...
	SolidWorks.Interop.sldworks.dll	E:\Work\001- Freelanc...

Name:
SolidWorks.Interop.swcommands.dll
Created by:

File Version:
30.1.0.82

- From the appeared window press"Browse" and head to SOLIDWORKS location in program file directory.
- Choose the following files and press Ok:

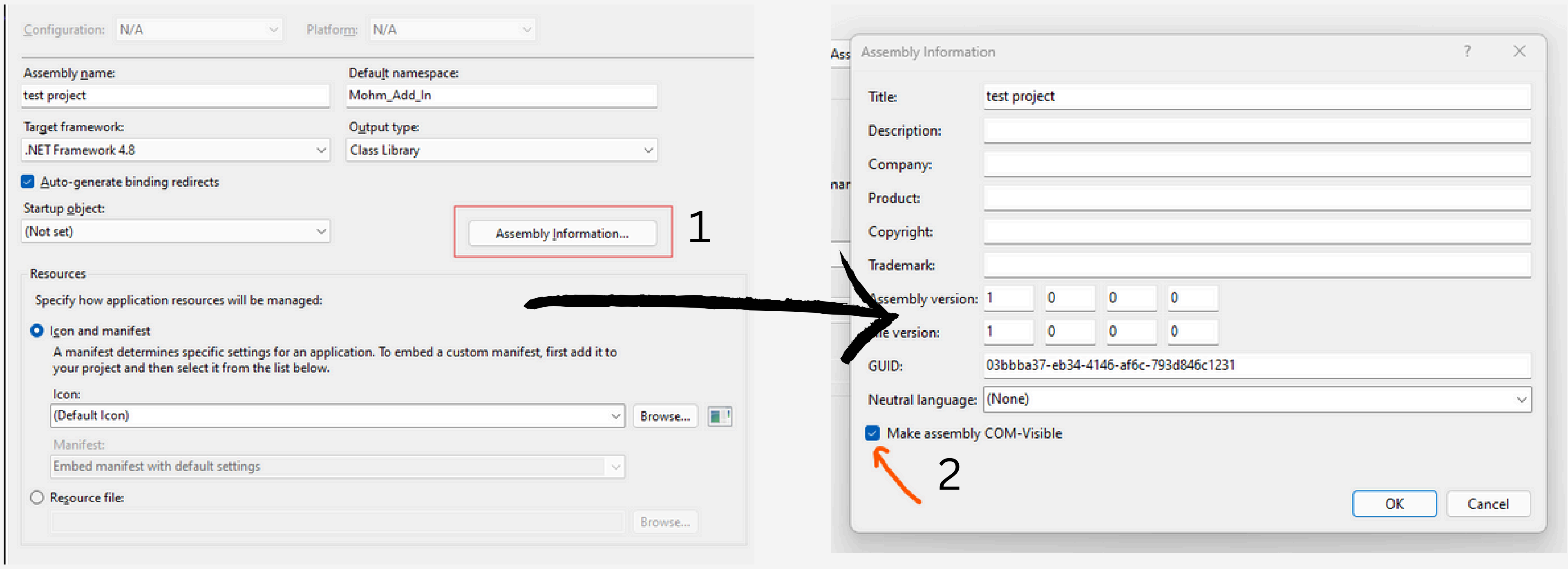
SolidWorks.Interop.sldworks.dll
SolidWorks.Interop.swcommands.dll
SolidWorks.Interop.swconst.dll
SolidWorks.Interop.swdimxpert.dll
SolidWorks.Interop.swdocumentmgr.dll
SolidWorks.Interop.swmotionstudy.dll
SolidWorks.Interop.swpublished.dll

Ahmed Hamed
SOLIDWORKS Expert
info@aterosolutions.com

#03 PREPARE PROJECT (1)

There is some steps you need to do in order to make your project run the code smoothly without nay problems.

- Select the project name and press right click, and choose "Properties", follow the next screens, and make sure to tick "Make assembly COM-Visible"



Configuration: N/A
Platform: N/A

Assembly name: test project
Default namespace: Mohm_Add_In
Target framework: .NET Framework 4.8
Output type: Class Library
☑ Auto-generate binding redirects
Startup object: (Not set)

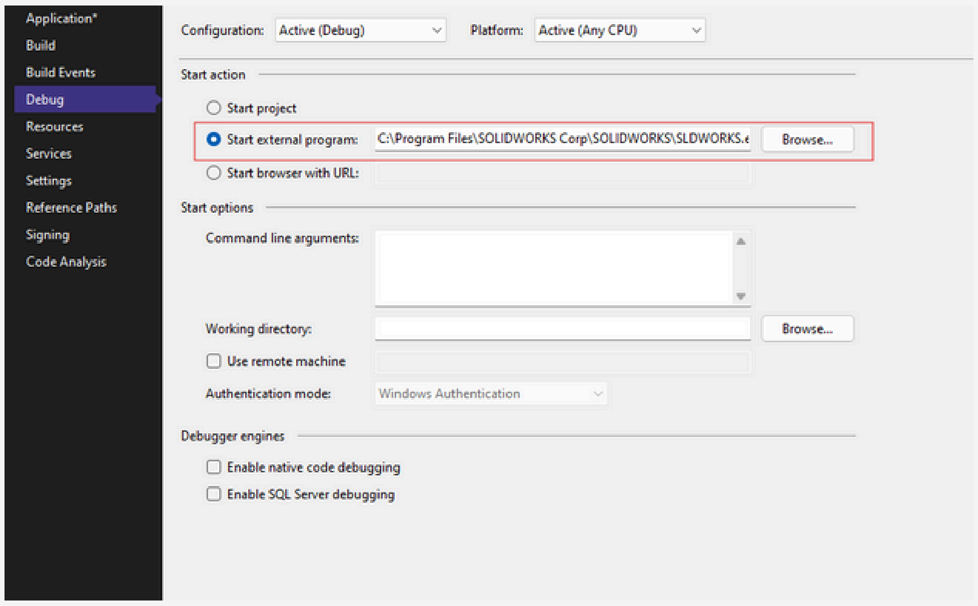
Resources
Specify how application resources will be managed:
● Icon and manifest
A manifest determines specific settings for an application. To embed a custom manifest, first add it to your project and then select it from the list below.
Icon: (Default Icon)
Manifest: Embed manifest with default settings
○ Resource file:

Assembly Information

Title:	test project			
Description:				
Company:				
Product:				
Copyright:				
Trademark:				
Assembly version:	1	0	0	0
File version:	1	0	0	0
GUID:	03bbba37-eb34-4146-af6c-793d846c1231			
Neutral language:	(None)			

☑ Make assembly COM-Visible

- While still in Option, choose debug, and click BROWSE, then head to"SLDWORKS.exe" in the main SOLIDWORKS folder in Program Files



Configuration: Active (Debug)
Platform: Active (Any CPU)

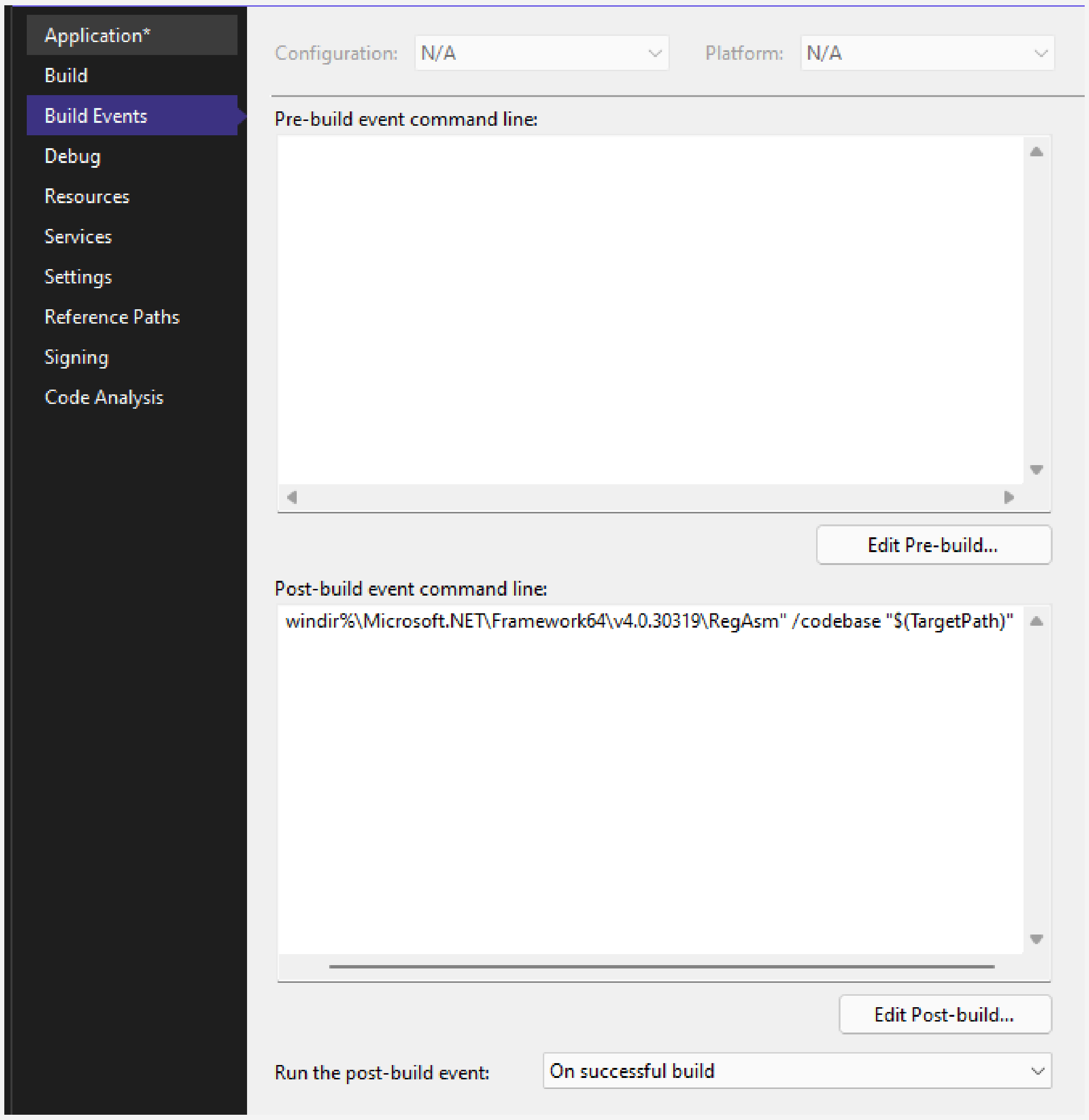
Start action
○ Start project
● Start external program: C:\Program Files\SOLIDWORKS Corp\SOLIDWORKS\SLDWORKS.e
○ Start browser with URL:

Start options
Command line arguments:
Working directory:
□ Use remote machine
Authentication mode: Windows Authentication

Debugger engines
□ Enable native code debugging
□ Enable SQL Server debugging

#03 PREPARE PROJECT (2)

- Finally, Select “Build Events”



Configuration: N/A Platform: N/A

Pre-build event command line:

Post-build event command line:

```
windir%\Microsoft.NET\Framework64\v4.0.30319\RegAsm" /codebase "$(TargetPath)"
```

Run the post-build event: On successful build

- In Post-build event command line, type the following code:

```
"%windir%\Microsoft.NET\Framework64\v4.0.30319\RegAsm" /codebase "$(TargetPath)"
```

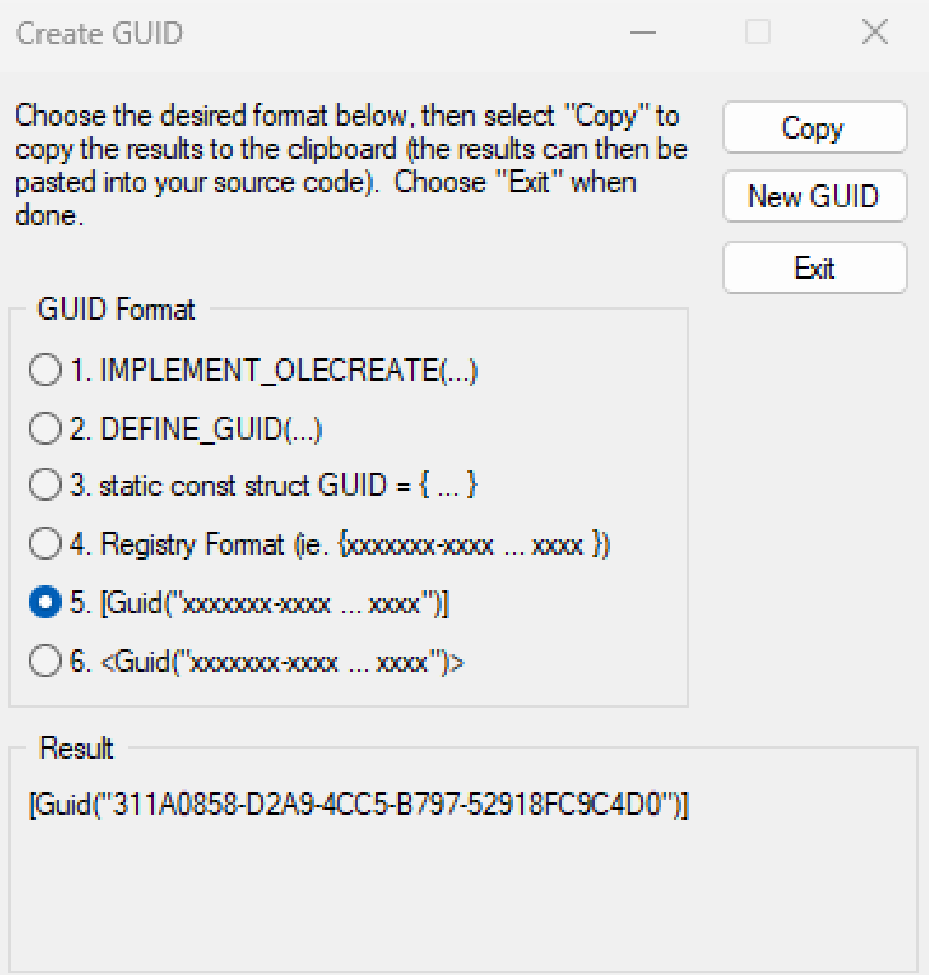
Now You are ready for writing your code 🙂

Ahmed Hamed
SOLIDWORKS Expert
info@aterosolutions.com

#04 PREPARE FOR CODING

- It is better to rename “class 1” to something more readable, let’s rename it “TaskPaneIntegration”, and make sure to press yes to the message that tells you that all the relative objects will be renamed as well.

- You will need a unique GUID for the registry. To get one from visual studio press on Tools > Create GUID, and from the pop up windows copy your GUID.



Create GUID

Choose the desired format below, then select "Copy" to copy the results to the clipboard (the results can then be pasted into your source code). Choose "Exit" when done.

GUID Format

○ 1. IMPLEMENT_OLECREATE(...)
○ 2. DEFINE_GUID(...)
○ 3. static const struct GUID = { ... }
○ 4. Registry Format (ie. {xxxxxxx-xxxx ... xxxx })
● 5. [Guid("xxxxxxx-xxxx ... xxxx")]
○ 6. <Guid("xxxxxxx-xxxx ... xxxx")>

Result

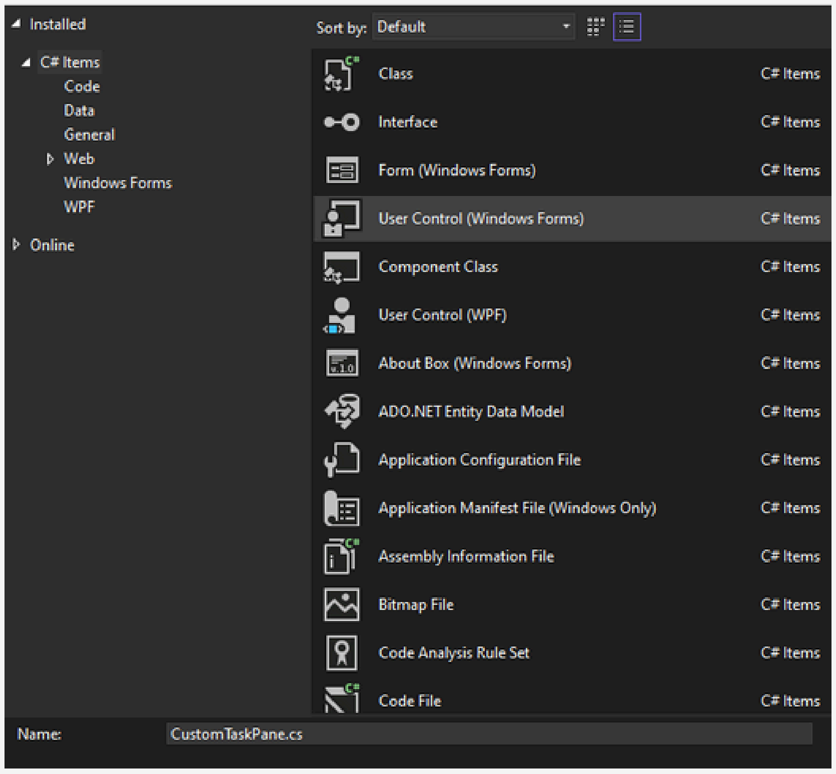
[Guid("311A0858-D2A9-4CC5-B797-52918FC9C4D0")]

- To make your add-in has its own picture in the task pane, in the main folder of your project, put the photo you want.

Ahmed Hamed
SOLIDWORKS Expert
info@aterosolutions.com

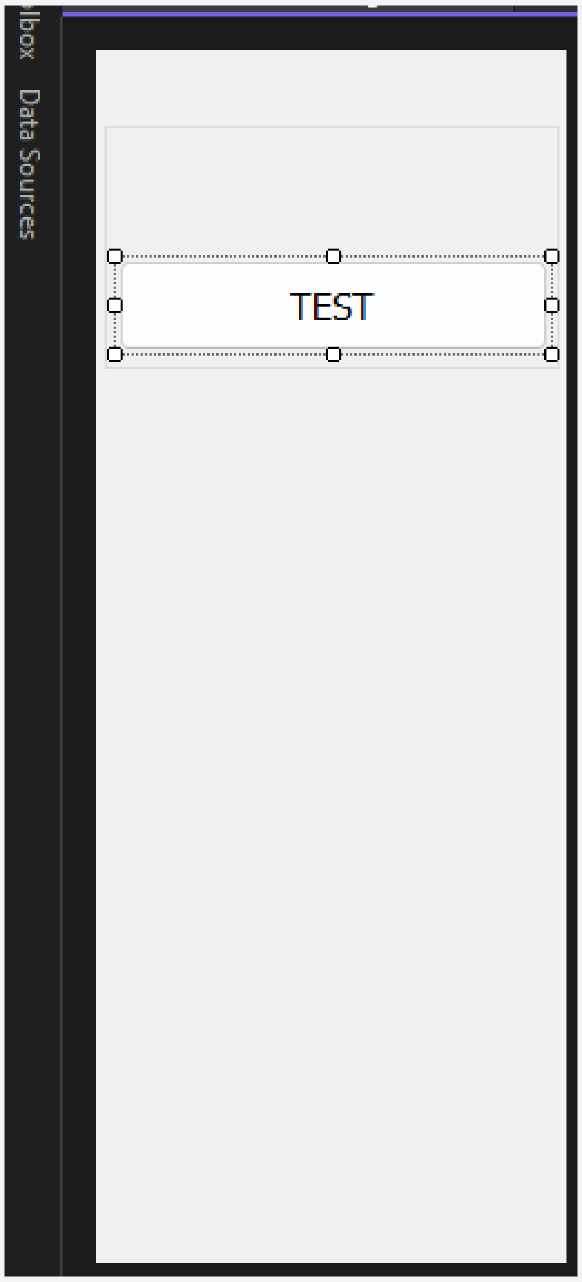
#05 DESIGN THE TASK PANE

- Right click on the project name on the right panel and choose Add > User control.
- from the appeared window choose “User Control” , and name it “CustomTaskPane”, Then press Add



Sort by: Default

Name: CustomTaskPane.cs



- After that, in the right side, right click on the “CustomTaskPane” and choose “View Designer”.
- From there the design of the task pane will appear, you can add buttons, text, labels, and all the tools you normally assign in your visual studio projects.
- you can continue to create your custom add-in function, then come back to complete adding it to task pane, or vise versa.

#06 START CODING

- Right click on “TaskPaneIntegration.cs” file in the right section , and choose “View Code”.
- In the code area, paste the following code, make sure that all the code will be inside the “namespace” section.

```
[ComVisible(true)]
[Guid("A418D0B5-FB95-4E5A-B842-8137D0B4CF45")]

public class TaskPaneIntegration : ISwAddin

{
 private int InnwCookie;
 private TaskpaneView CustomTaskpaneView;
 private CustomTaskPane TaskPaneHost;
  public static SldWorks SolidworksApplication;
 public const string swProgID = "Enable Addin-For-Solidworks";
 public static string workPath =
Path.GetDirectoryName(typeof(TaskPaneIntegration).Assembly.CodeBase).Replace(@"file:\", string.Empty);
 public bool ConnectToSW(object ThisSW, int Cookie)
 {
 }
 public bool DisconnectFromSW()
 {
 }
 private void LoadUI()
 {
 }
 private void UnLoadUI()
 {
 }
 [ComRegisterFunction()]
 private static void ComRegister(Type t)
 {
 }
 [ComUnregisterFunction()]
 private static void ComUnRegister(Type t)
 {
 }

}
```

Inside this class we will write all the variables deceleration, and our 6 needed methods

Make sure to change the GUID in this code with the one you already have.

Ahmed Hamed
SOLIDWORKS Expert
info@aterosolutions.com

#07 FILL THE METHODS (1)

Now, fill the 6 methods as following.

```
public bool ConnectToSW(object ThisSW, int Cookie)
{
 SolidworksApplication = (SldWorks)ThisSW;
 InnwCookie = Cookie;
 var ok = SolidworksApplication.SetAddinCallbackInfo2(0, this, InnwCookie);
 LoadUI();
 return true;
}
```

```
public bool DisconnectFromSW()
{
 UnLoadUI();
 return true;
}
```

```
private void LoadUI()
{
 // Find location to our taskpane image
 var imagePath =
Path.Combine(Path.GetDirectoryName(typeof(TaskPaneIntegration).Assembly.CodeBase).Replace(@"file:\",
string.Empty), "Logo.png");

 // Craete our Taskpane
 CustomTaskpaneView = SolidworksApplication.CreateTaskpaneView2(imagePath, "custom Add-in");

 // Load our UI into our Taskpane

 TaskPaneHost = (CustomTaskPane)CustomTaskpaneView.AddControl(TaskPaneIntegration.swProgID,
string.Empty);

}
```

In the previous method, make sure to write the name of the picture of your add-in as it exist in the main folder.

#07 FILL THE METHODS (2)

Continue to fill the 6 methods as following.

```
private void UnLoadUI()
{
 TaskPaneHost = null;


 // Remove Taskpane view
 CustomTaskpaneView.DeleteView();

 // Release COM reference and cleanup memory
 Marshal.ReleaseComObject(CustomTaskpaneView);

 //
 CustomTaskpaneView = null;
}
```

```
[ComRegisterFunction()]
private static void ComRegister(Type t)
{
 var keyPath = string.Format(@"SOFTWARE\Solidworks\AddIns\{0:b}", t.GUID);

 using (var rk = Microsoft.Win32.Registry.LocalMachine.CreateSubKey(keyPath))
 {
  // Load Addin when solidworks start
  rk.SetValue(null, 1);

  rk.SetValue("Title", "custom Add-in");
  rk.SetValue("Description", "For any help info@aterosolutions.com");
 }

 var keyPath2 = string.Format(@"Software\SolidWorks\AddInsStartup\{0:b}", t.GUID);

 using (var rk2 = Microsoft.Win32.Registry.CurrentUser.CreateSubKey(keyPath2))
 {
  rk2.SetValue(null, 1);
 }

}
```

Ahmed Hamed
SOLIDWORKS Expert
info@aterosolutions.com

#07 FILL THE METHODS (3)

Continue to fill the 6 methods as following.

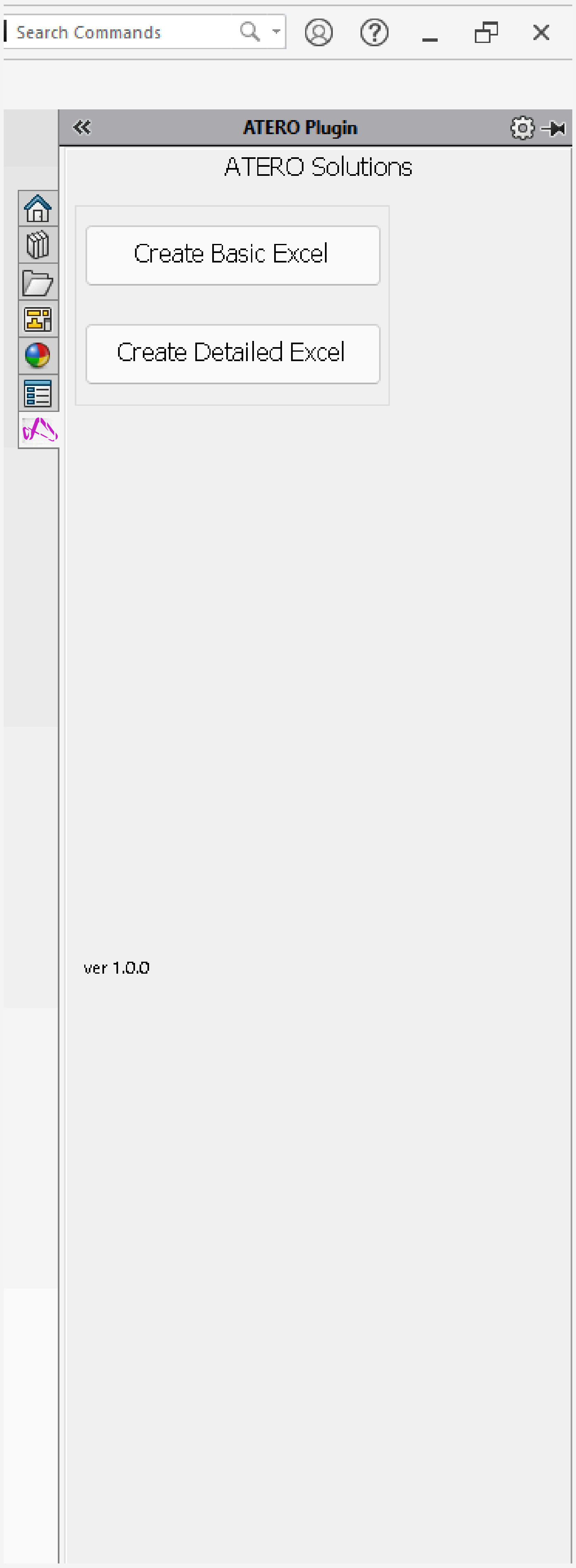
```
[ComUnregisterFunction()]
private static void ComUnRegister(Type t)
{
 var keyPath = string.Format(@"SOFTWARE\Solidworks\AddIns\{0:b}", t.GUID);

 Microsoft.Win32.Registry.LocalMachine.DeleteSubKeyTree(keyPath);

 var keyPath2 = string.Format(@"Software\SolidWorks\AddInsStartup\{0:b}", t.GUID);

 Microsoft.Win32.Registry.CurrentUser.DeleteSubKeyTree(keyPath2);
}
```

Ahmed Hamed
SOLIDWORKS Expert
info@aterosolutions.com

#08 RUN YOUR PROJECT

Congratulation. You have created your custom add-in and attached it to the task pane.

- In order to run your code, make sure that you open Visual Studio as administrator, and there is no SOLIDWORKS instances are opened.
- Press “Start” in the standard tool bar, you will see that a new instance of solidworks is now opening, and after that you will see anew tab in the task pane, with the custom photo you have put.
- press on that new tab, and you will see the task pane design you have created, and you can use it.
- This procedure has to be done one time only, every time you open SOLIDWORKS again you well see your custom add-in exist.



Ahmed Hamed
SOLIDWORKS Expert
info@aterosolutions.com

SOLIDWORKS

ATERO Solutions

Mastering SOLIDWORKS API is a huge thing, but it seems very easy if you are following us.

We have a numerous number of experts in SOLIDWORKS, in addition to other software.

Always be prepared for every guide and everything new from us

http:// www.aterosolutions.com