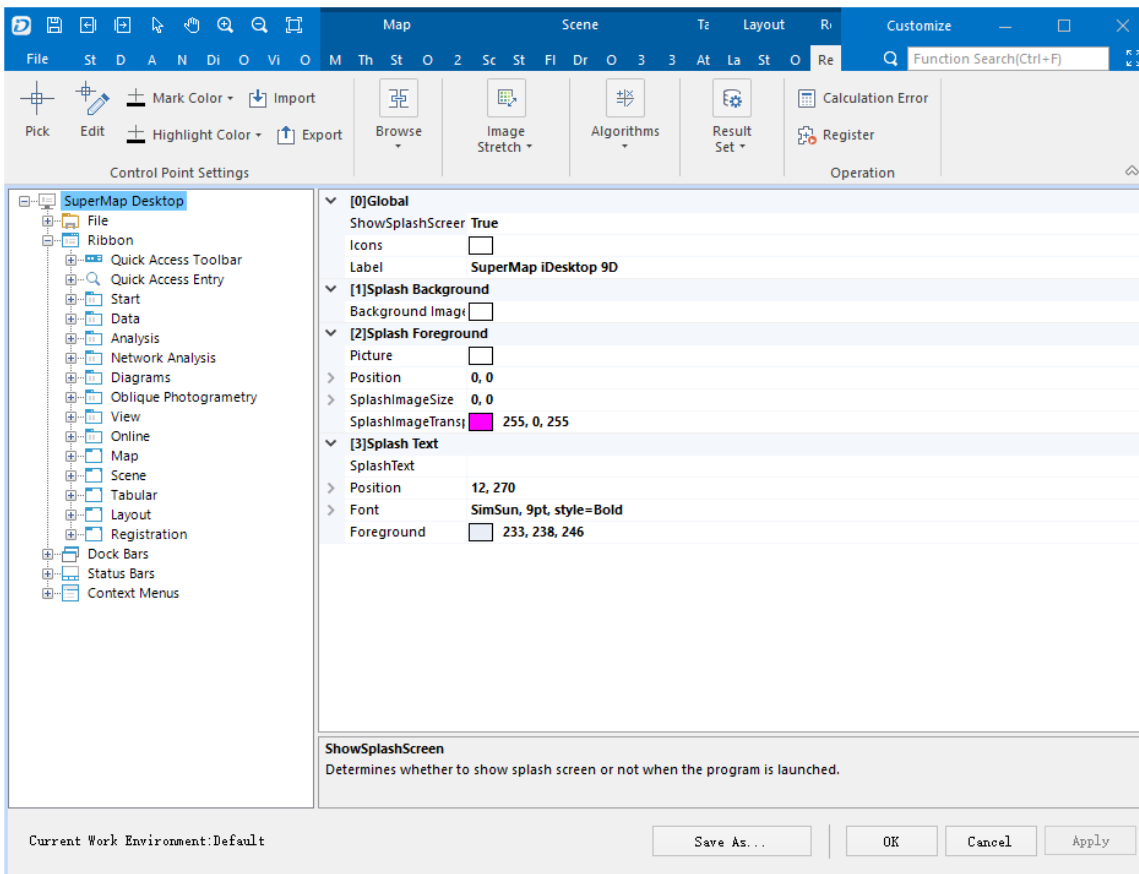# Work Environment Design

Note: SuperMap iDesktop 8C (2017) version of the working environment can not be temporarily designed, follow-up version will open this feature.

## Main Contents:

Step 1: Opening the Customize Window

Objective: Open the Customize window for work environment customization.

1. Start SuperMap iDesktop.
2. Click View > Custom Interface > Design > Customize to open the Customize window.
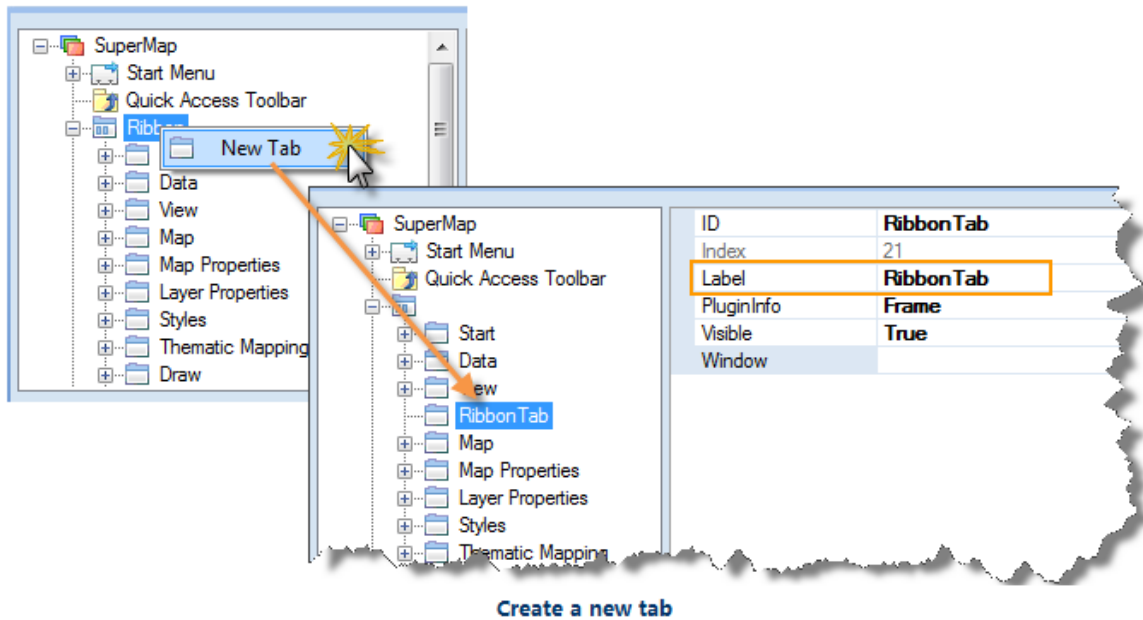


Open the Customize window

In the Customize window, you can add or delete controls and configure actions for controls.

This part introduces how to achieve customization and extension development by adding tabs, groups, buttons and writing code.

## Step 2: Adding a Tab

Objective: Add a new tab with the name "My Tools" to the Ribbon.

1. In the structure tree on the right side of the Customize window, right click the Ribbon node and click New Tab to create a new tab with the default name "RibbonTab".
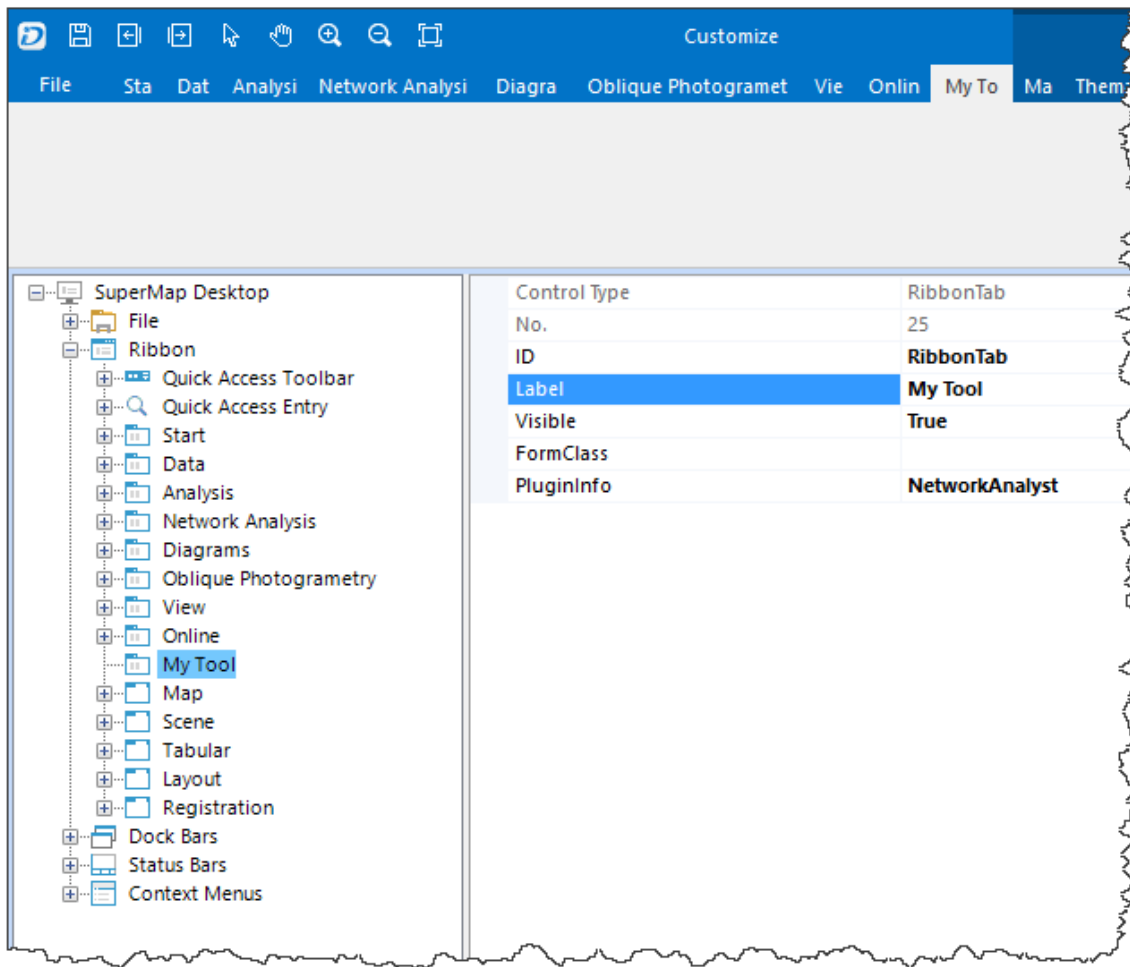


Create a new tab

2. Change the name of the new tab to "My Tools".

A blank My Tools tab shows up.

3. Adjust the position of the tab.

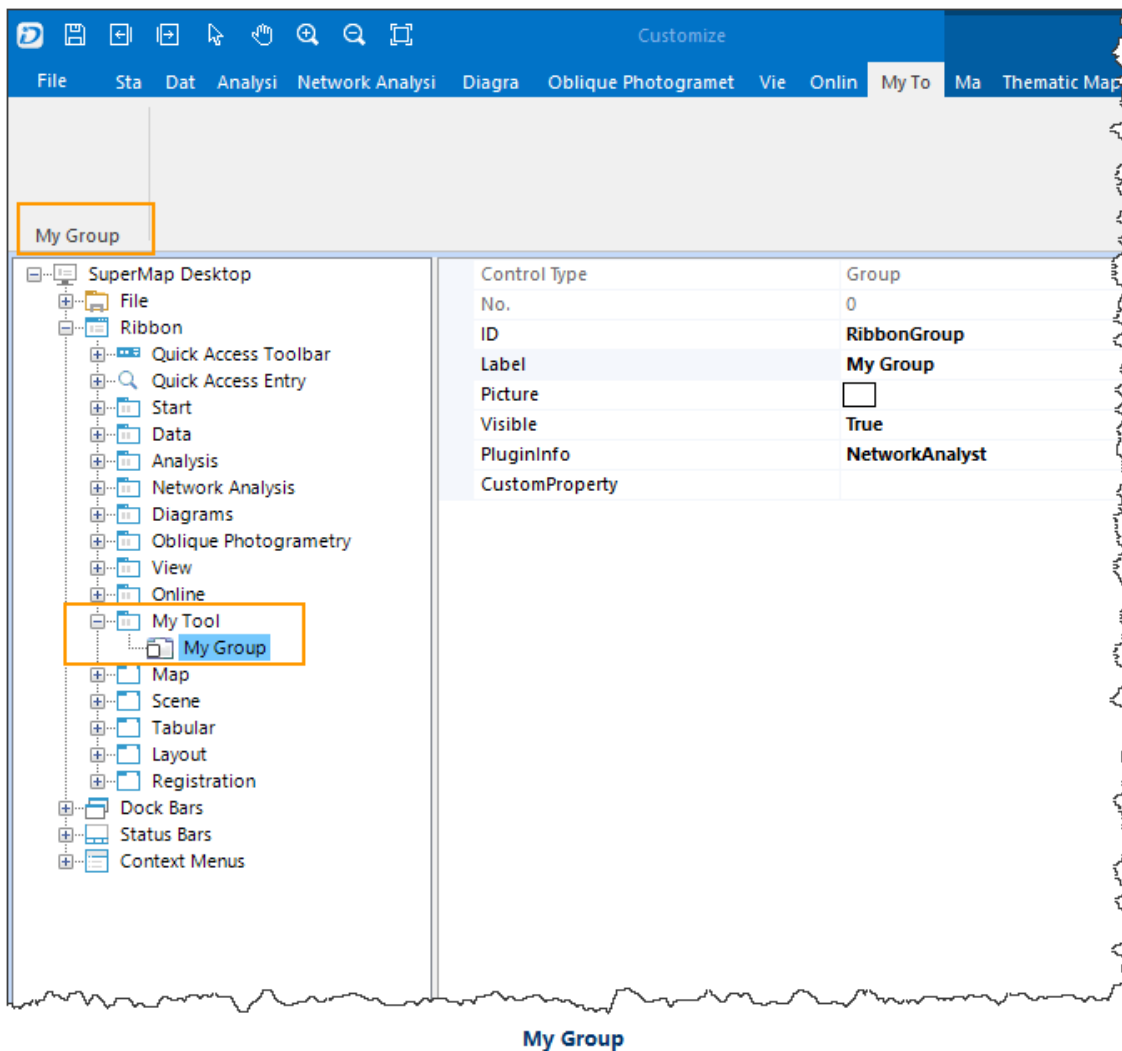Click the My Tools tab and drag it to the left of the Start tab.

**Adjust the position of the tab**

## Step 3: Adding a Group

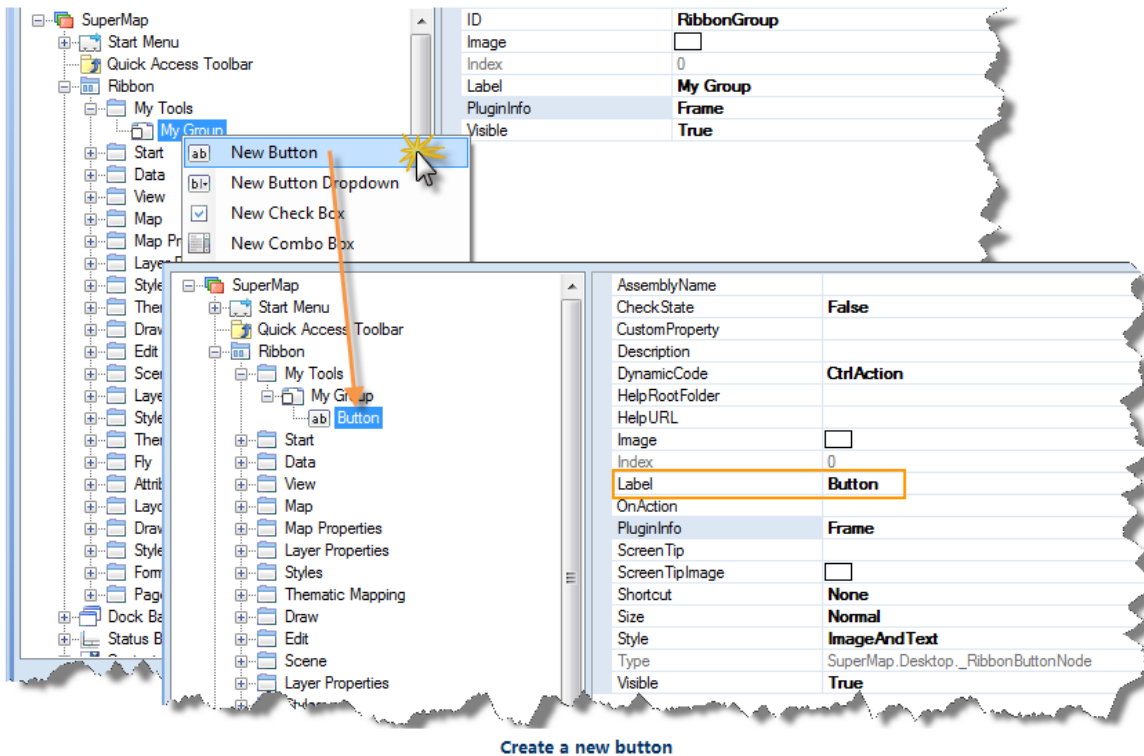Objective: Add a new group with the name "My Group" to My Tools.

1. Under the Ribbon node, right click My Tools and click New Group to create a new group with the default name "RibbonGroup".
2. Change the name of the new group to "My Group".

**My Group**

## Step 4: Adding a Button

Objective: Add a new button with the name "Plugin Info" to My Group.

1. Under the My Tools node, right click My Group and click New Button to create a new button with the default name "Button".
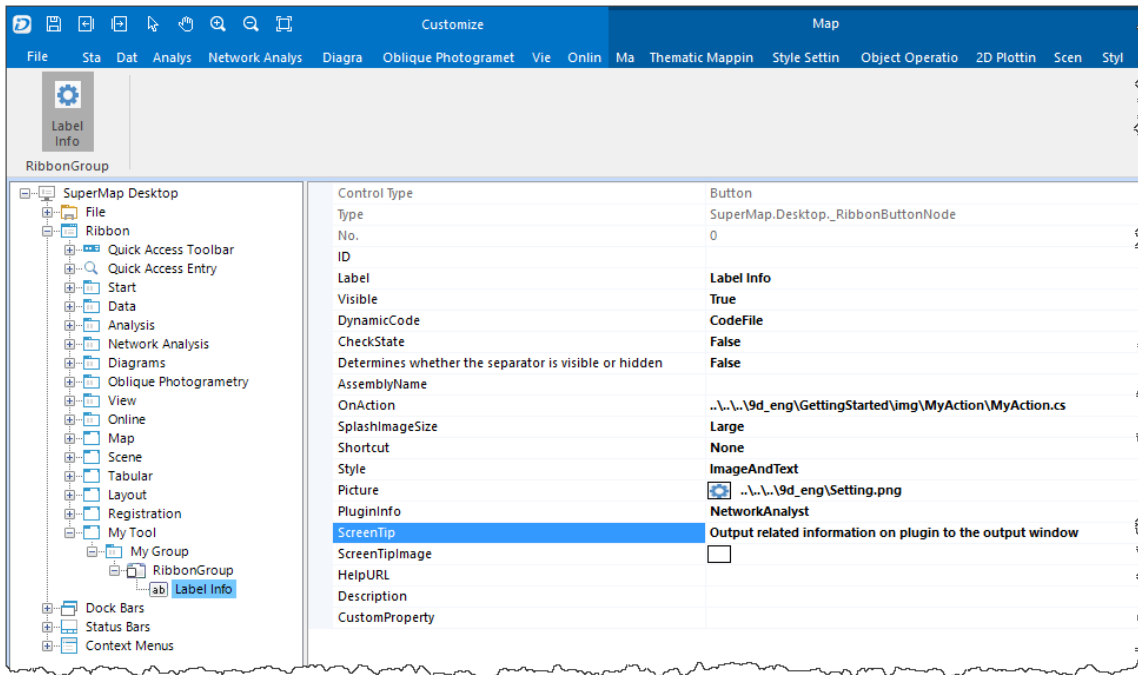
Create a new button

2. Change the properties of the new button.

| Property | Value | Description |
|---|---|---|
| CheckState | False | Indicates whether the button is checked by default. |
| Image | Plugin.png | Specifies the full path of the image you want to display on the button. Both an absolute path or a path relative to the executable program are accepted. Here you can right click Plugin.png, then click Save Target As to download the sample image. |
| Label | Plugin Info | The text associated with the button. |
| ScreenTip | Output information about all plugins in the Output Window. | Specifies the prompt message displayed when the mouse is moved over the button. The SuperTip mode is supported. |
| Size | Large | Specifies the display size of the button. Normal: 16*16，Large; Large: 32*32. |
| Style | ImageAndText | The display style of the button. Text: displays button label only; Image: dispplays button icon only; ImageAndText: displays both button name and icon. |
| Visible | True | Determines whether the button is visible or hidden. |

After changing the properties, the button is shown as follows:

**Plugin Info**

Step 5: Configuring Action

# Objectives:

- Write code to achieve the traversal of all plugins and information output.
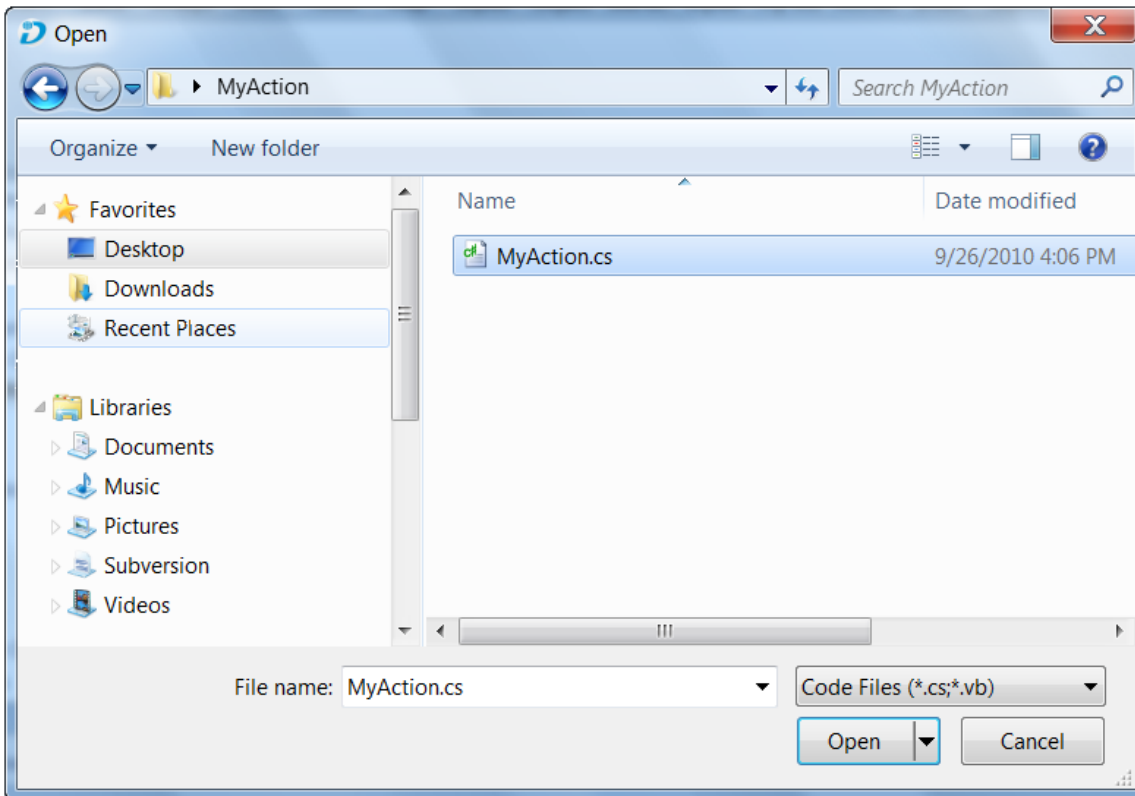- Configure the function for the Plugin Info button.

# Realization:

There are three modes to set the DynamicCode for a button: CtrlAction, ScriptCode and CodeFile.
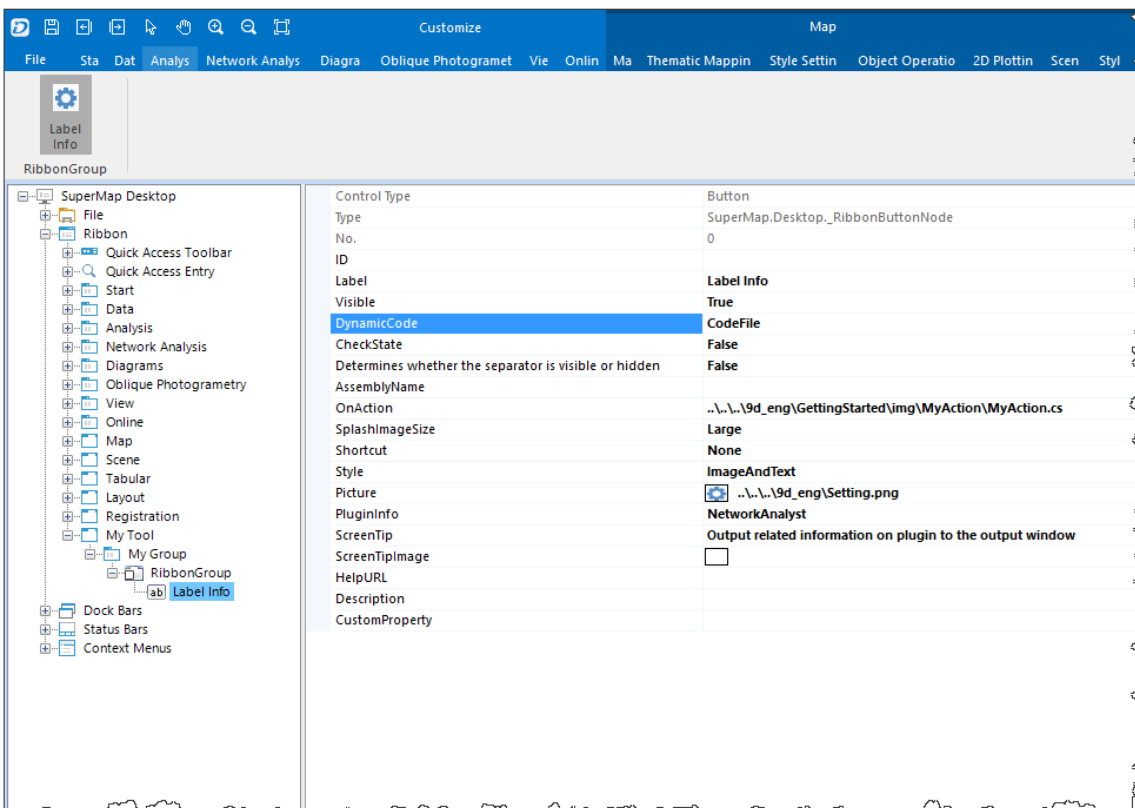
The three modes are:

Mode 1: CodeFile

1. Right click MyAction.rar and click Save Target As to download MyAction.cs to the local.
2. Set the DynamicCode property to CodeFile. In the Open dialog box that shows up, navigate to MyAction.cs, highlight it and click Open.
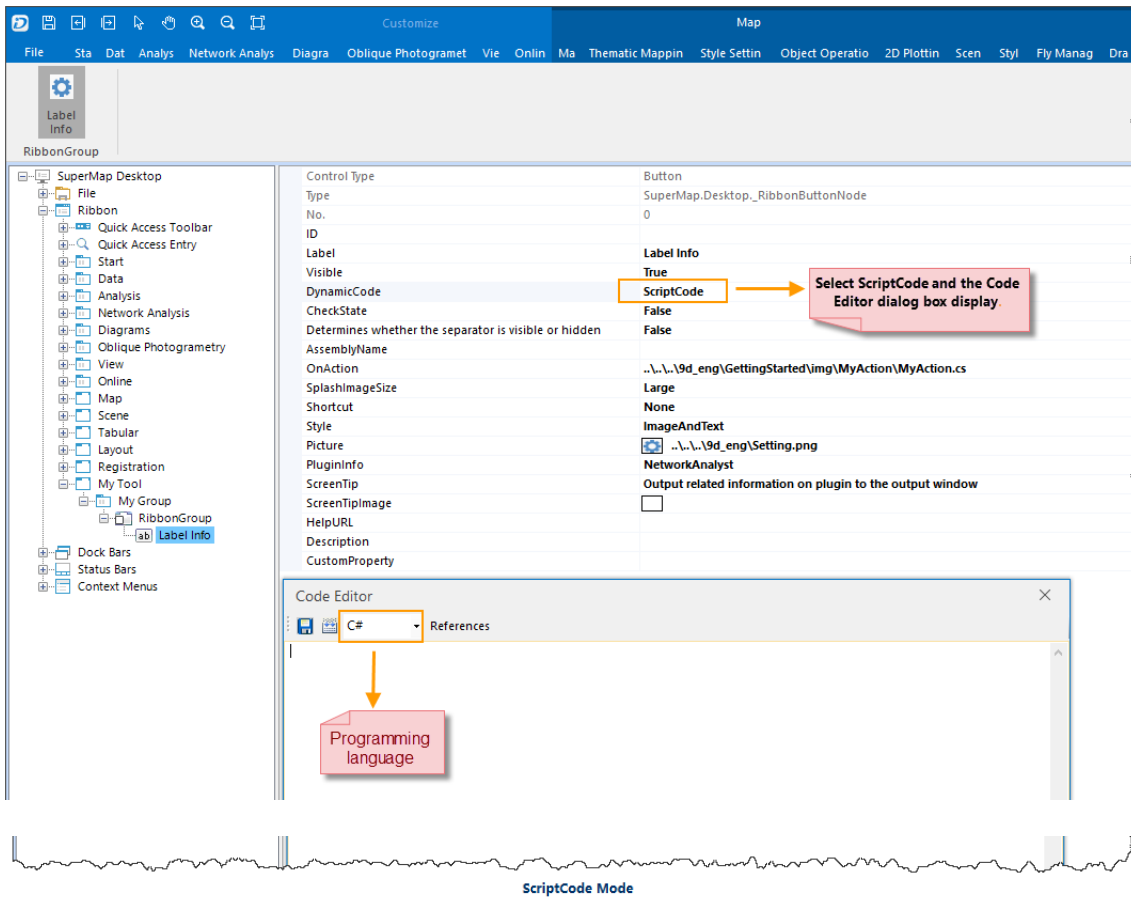
**Open dialog box**

3. When finished, click OK to save the settings.



**Properties of the Plugin Info button**

Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com      Website: www.supermap.com

## Mode 2: ScriptCode

1. Set the DynamicCode property to ScriptCode. The Script Complier window shows up.



**ScriptCode Mode**

2. Choose C# and copy the code below to the Script Compiler window.

```
//Get the total number of plugins in the application program
Int32 pluginCount = SuperMap.Desktop.Application.ActiveApplication.PluginManager.Count;

//Traverse all plugins and output plugins information in the Output Window
for (Int32 index = 0; index < pluginCount; index++)
{
    //Get the class object that defines the plugin

    Plugin plugin = SuperMap.Desktop.Application.ActiveApplication.PluginManager[index];

    //Get the info object of the plugin

    PluginInfo pluginInfo = plugin.PluginInfo;

    //Get the name of the plugin

    String pluginName = pluginInfo.Name;

    //Get the full path of the assembly where the plugin is located

    String pluginAssemble = pluginInfo.AssemblyName;

    //Output the plugin name and the assembly to which the plugin belongs in the Output Window

    SuperMap.Desktop.Application.ActiveApplication.Output.Output("Plugin Name: " + pluginName + "\r\n" + "Assembly: " + pluginAssemble);
}
```
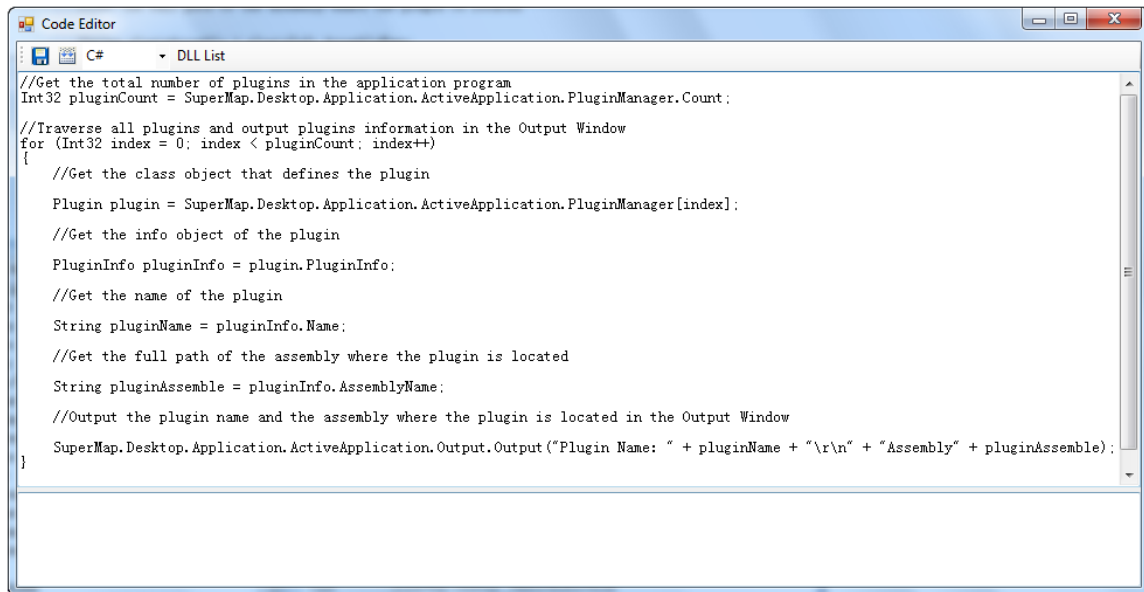
3. Click Build. When succeeded, click Save.

Close the Script Compiler window.

```
Code Editor

//Get the total number of plugins in the application program
Int32 pluginCount = SuperMap.Desktop.Application.ActiveApplication.PluginManager.Count;

//Traverse all plugins and output plugins information in the Output Window
for (Int32 index = 0; index < pluginCount; index++)
{
    //Get the class object that defines the plugin

    Plugin plugin = SuperMap.Desktop.Application.ActiveApplication.PluginManager[index];

    //Get the info object of the plugin

    PluginInfo pluginInfo = plugin.PluginInfo;

    //Get the name of the plugin

    String pluginName = pluginInfo.Name;

    //Get the full path of the assembly where the plugin is located

    String pluginAssemble = pluginInfo.AssemblyName;

    //Output the plugin name and the assembly where the plugin is located in the Output Window

    SuperMap.Desktop.Application.ActiveApplication.Output.Output("Plugin Name: " + pluginName + "\r\n" + "Assembly" + pluginAssemble);
}
```
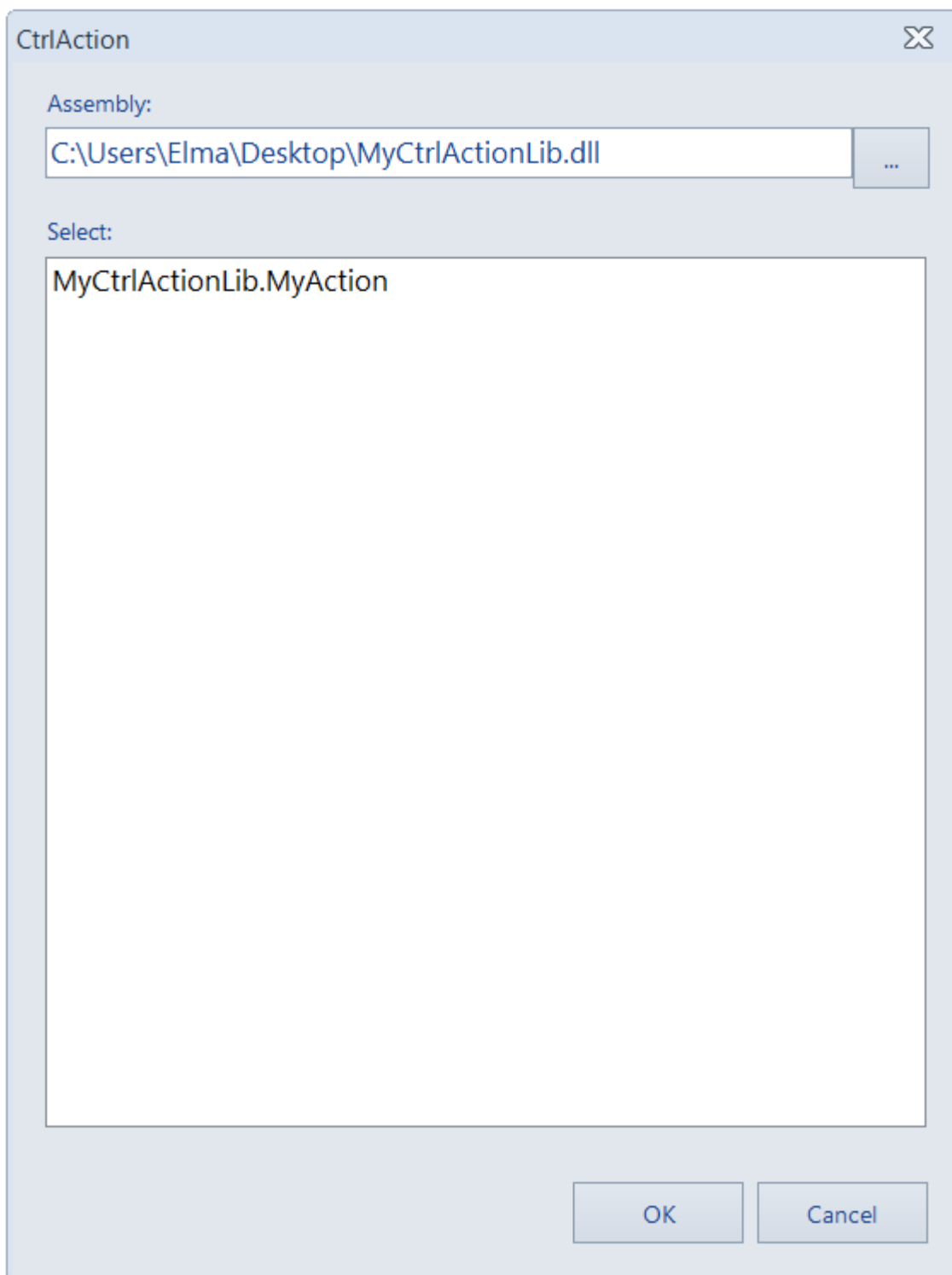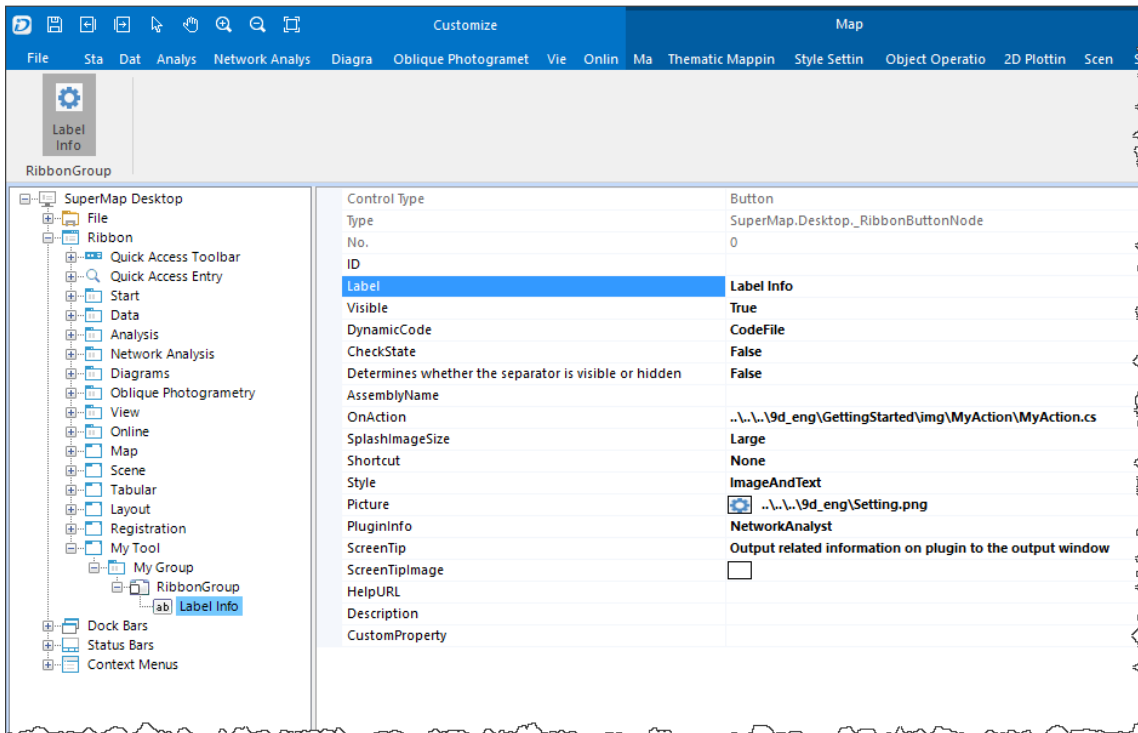
Save the script

4. Click OK to save the settings.

## Mode 3: CtrlAction

1. Right click MyCtrlAction.dll and click Save Target As to download MyCtrlActionLib.dll to the local.
2. Set the DynamicCode property to CtrlAction.
3. Click the button to the right of the OnAction property to display the CtrlAction dialog box. Click the button with the ellipse, navigate to MyCtrlActionLib.dll in the Open dialog box that shows up, and then click Open.
4. Select MyCtrlAction._CtrlAction in the class list and click OK.
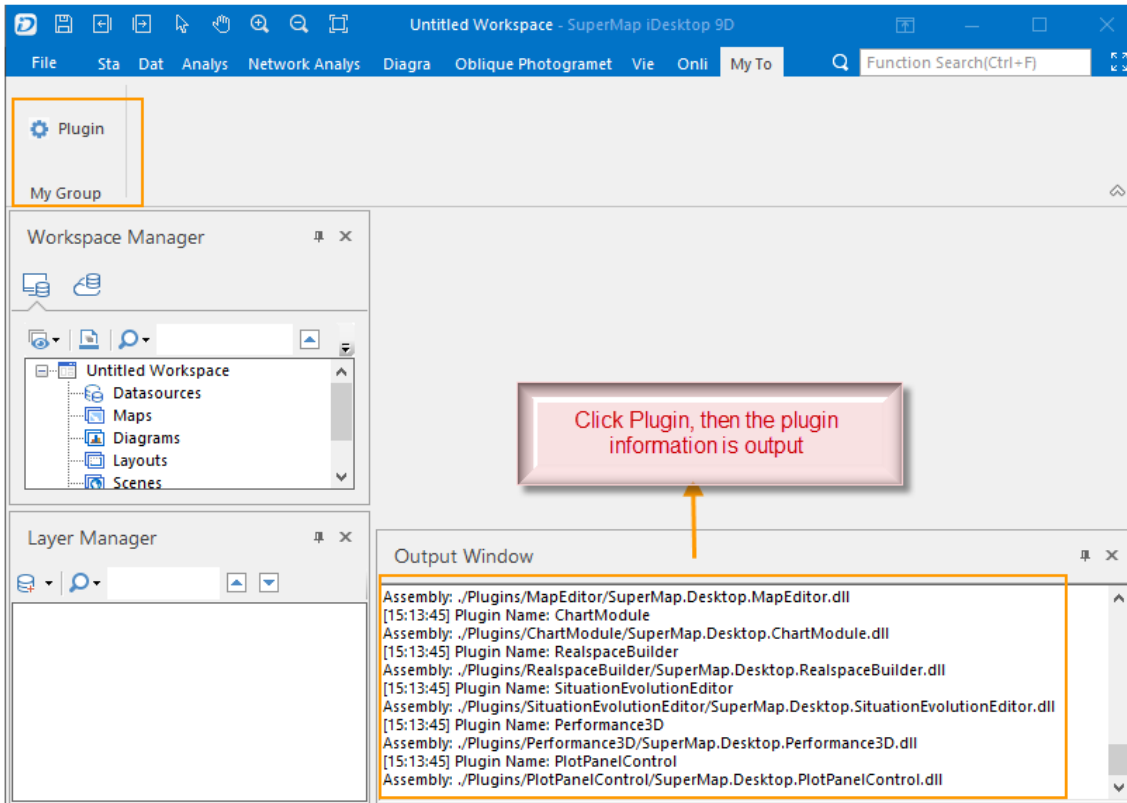
5. Click Ok to save the settings.

Properties of the Plugin Info button

## Note:

The action for a control event is specified by a class object implementing the ICtrlAction interface. When an event is triggered, the object's Run method, which overrides the Run method of the ICtrlAction interface, will be called. CtrlAction is a class that implements the ICtrlAction interface.

Step 6: Viewing Results

1. Click OK to save all settings and close the Customization window.
2. On the My Tools tab, in My Group, click Plugin Info. You will find the information about all plugins of the application program is output in the Output Window.

**Plugin Info**