

Walkthrough: Automatically Populating the Toolbox with Custom Components

.NET Framework 4.5 1 out of 2 rated this helpful

If your components are defined by a project in the currently open solution, they will automatically appear in the **Toolbox**, with no action required by you. You can also manually populate the **Toolbox** with your custom components by using the [Choose Toolbox Items Dialog Box \(Visual Studio\)](#), but the **Toolbox** takes account of items in your solution's build outputs with all the following characteristics:

- Implements [IComponent](#);
- Does not have [ToolboxItemAttribute](#) set to **false**;
- Does not have [DesignTimeVisibleAttribute](#) set to **false**.

Note

The **Toolbox** does not follow reference chains, so it will not display items that are not built by a project in your solution.

This walkthrough demonstrates how a custom component automatically appears in the **Toolbox** once the component is built. Tasks illustrated in this walkthrough include:

- Creating a Windows Forms project.
- Creating a custom component.
- Creating an instance of a custom component.
- Unloading and reloading a custom component.

When you are finished, you will see that the **Toolbox** is populated with a component that you have created.

Note

The dialog boxes and menu commands you see might differ from those described in Help depending on your active settings or edition. To change your settings, choose **Import and Export Settings** on the **Tools** menu. For more information, see [Customizing Development Settings](#).

Creating the Project

The first step is to create the project and to set up the form.

To create the project

1. Create a Windows-based application project called **ToolboxExample**.

For more information, see [How to: Create a New Windows Forms Application Project](#).

2. Add a new component to the project. Call it **DemoComponent**.

For more information, see [How to: Add New Project Items](#).

3. Build the project.

4. From the **Tools** menu, click the **Options** item. Click **General** under the **Windows Forms Designer** item and ensure that the **AutoToolboxPopulate** option is set to **True**.

Creating an Instance of a Custom Component

The next step is to create an instance of the custom component on the form. Because the **Toolbox** automatically accounts for the new component, this is as easy as creating any other component or control.

To create an instance of a custom component

1. Open the project's form in the **Forms Designer**.

2. In the **Toolbox**, click the new tab called **ToolboxExample Components**.

Once you click the tab, you will see **DemoComponent**.

Note

For performance reasons, components in the auto-populated area of the **Toolbox** do not display custom bitmaps, and the [ToolboxBitmapAttribute](#) is not supported. To display an icon for a custom component in the **Toolbox**, use the **Choose Toolbox Items** dialog box to load your component.

3. Drag your component onto your form.

An instance of the component is created and added to the **Component Tray**.

Unloading and Reloading a Custom Component

The **Toolbox** takes account of the components in each loaded project, and when a project is unloaded, it removes references to the project's components.

To experiment with the effect on the Toolbox of unloading and reloading components

1. Unload the project from the solution.

For more information about unloading projects, see [How to: Unload and Reload Projects](#). If you are prompted to save, choose **Yes**.

2. Add a new **Windows Application** project to the solution. Open the form in the **Designer**.

The **ToolboxExample Components** tab from the previous project is now gone.

3. Reload the **ToolboxExample** project.

The **ToolboxExample Components** tab now reappears.

Next Steps

This walkthrough demonstrates that the **Toolbox** takes account of a project's components, but the **Toolbox** is also takes account of controls. Experiment with your own custom controls by adding and removing control projects from your solution.

See Also

Reference

[General, Windows Forms Designer, Options Dialog Box](#)

Other Resources

[How to: Manipulate Toolbox Tabs](#)

[Choose Toolbox Items Dialog Box \(Visual Studio\)](#)

[Putting Controls on Windows Forms](#)

Was this page helpful? ☐ Yes ☐ No