

Customizing GUI Components

This document describes how to create, customize, and reuse GUI components in TIBCO General Interface Builder. The use of custom components offers many advantages in application development — component reusability, consistency of look and feel across applications, faster development time, and reduction in repetitive, time-consuming component development.

After you've created customized components, you can add them to your user component library and access them from any project, as well as share them with the development team.

Version 1.0. 05-16-2006

Scope: Version 3.1.1



<http://www.tibco.com>

Global Headquarters

3303 Hillview Avenue

Palo Alto, CA 94304

Tel: +1 650-846-1000

Toll Free: 1 800-420-8450

Fax: +1 650-846-1005

© 2006, TIBCO Software Inc. All rights reserved. TIBCO, the TIBCO logo, The Power of Now, and TIBCO Software are trademarks or registered trademarks of TIBCO Software Inc. in the United States and/or other countries. All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

Table of Contents

Overview	3
Creating a Custom Button	3
Saving a Custom Button.....	3
Using a Custom Button.....	4

Overview

In this document, you will learn how to create a custom button component that is independent of any project. The use of custom components offers many advantages in application development:

- Component reusability
- Consistent look and feel across applications
- Faster development time
- Reduced rework of components

Once you've created a custom component, you can use it as is in your application or modify its properties. In addition, you can share it across the development team.

In this example, you'll create a custom button, save it in a component library, and then use it in your application.


Creating a Custom Button

Let's begin by creating a custom button as follows:

1. Open or create a project in General Interface Builder.
2. Press **Ctrl+n** or choose **File > New > GUI Component** to create a new component.
3. Drag and drop a button from **System > Form Elements** in the Component Library palette to the work area.
4. Open the **Events Editor palette** and clear the alert value for the Execute event. Typically, you won't want to assign an event until you use it in your application.
5. Open the **Properties Editor palette** and enter the following property values to give the button a custom look-and-feel:

Property Name	Property Value
- Color	blue
- Bgcolor	#ccffcc
- Font Weight	Bold

Note: You can enter your own hexadecimal value or use the Color Picker to choose a color value (Tools > Color Picker).

6. Click the **Component Profile button**  at the bottom right of the work area.
 - a. Enter `Blue Button` in the Name field. This is the name that displays in the Component Library palette.
 - b. Enter a description. This information displays in a tooltip when you hover over the component name in the Component Libraries palette.
 - c. Type in the URL for an optional custom image. For example, `user\images\icon1.gif`.
Note: If you're using images across your applications, you might create an `images` folder for all your images in the user directory.

Saving a Custom Button

After you've created the custom component, you need to save it to the `install_dir\user\prototypes` directory. Any components saved to this directory appear in the User folder of the Component Library palette. You can create nested folders to organize your components.

To save the custom button so that it will appear in the Component Library palette,

1. Press **Ctrl+s** to save the new component.
Navigate to the `install_dir\user\prototypes` folder.
2. Click the **Create New Folder** button in the Save File dialog.
3. Name the folder `Buttons` and click **Create**.
4. Navigate to the **Buttons** folder.
5. Save the file with the name `BlueButton.xml`.
6. Click the **Reload** button on the Component Libraries palette toolbar to load the new component from disk.

7. Open the **User > Buttons** folders in the Component Library palette to see the Blue Button you just created. If you entered an image URL, you should also see an icon for the button. Move the mouse over the button to see the description in a tooltip.

Using a Custom Button

Now that you've created the custom button, you can use it in your application just as you would with any of the pre-built GUI components in the Component Libraries palette. You can also modify the button as needed once you drop it on your layout.

To use and modify your custom button, complete the following steps:

1. Create or open a project.
2. Drag and drop a **Block** component onto `appCanvas.xml`.
3. Drag and drop **Blue Button** from **User > Buttons** in the Component Library palette to `appCanvas.xml`.
4. Modify the properties of the button if needed or use it as is.
5. Add an Execute function in the `Events Editor` palette.
6. Right-click the `appCanvas.xml` tab and select **Save and Reload**.

Now that you have a custom button, you can continue to develop your own library of custom components. Simply choose a component in the Component Library palette, modify it, and save it to the `user\prototypes` folder to add it to your User library. As you can see, using custom components can save time by eliminating repetitive rework and can also facilitate creating a common look-and-feel across the application.