

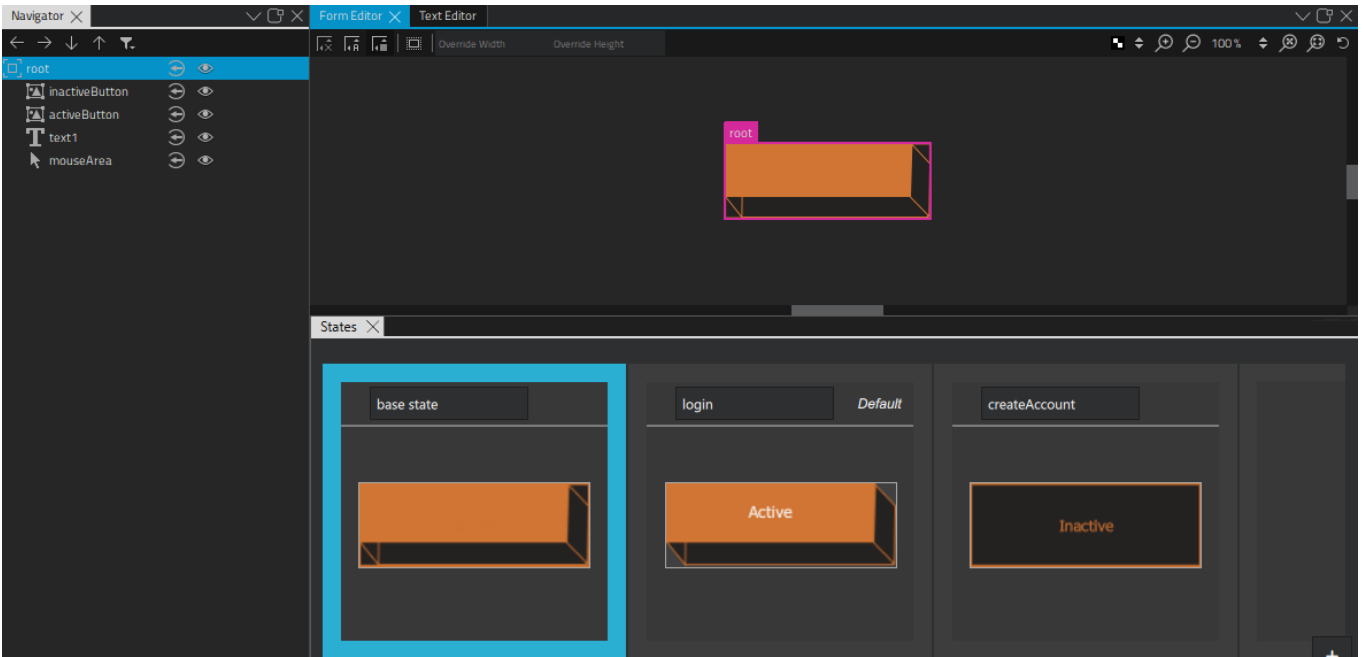
# 创建可缩放的按钮和边框

可以使用“[边框图像](#)”组件将图像（如 PNG 文件）显示为边框和背景。

使用两个边框图像和合适的图形来更改单击按钮时按钮的外观。可以使用使用[状态](#)来确定哪个图像可见，具体取决于是否按下了鼠标按钮。您可以添加更多图像和状态，以根据其他鼠标事件（如悬停）更改按钮的外观。

使用[文本](#)组件添加按钮文本。您还可以使用状态来更改按钮文本颜色和字体大小。例如，您可以向上或向下缩放按钮文本。

添加覆盖整个区域并对鼠标事件做出反应的鼠标[区域](#)组件。








## 创建按钮组件

若要创建按钮组件，请选择“**文件**”>“**新建文件**”>“**Qt 快速文件**”>“**Qt 快速 UI 文件**”>选择“以创建名为 Button.ui.qml 的 [UI 文件](#)（例如）。

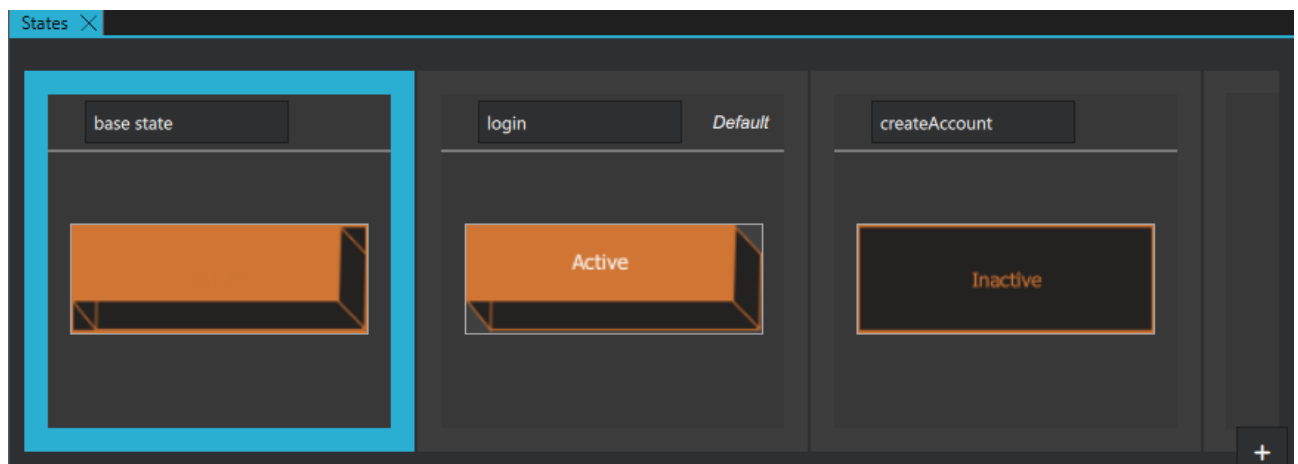
**注意：** 仅当文件名以大写字母开头时，组件>“**我的组件**”中才会列出组件。


## 构造按钮组件

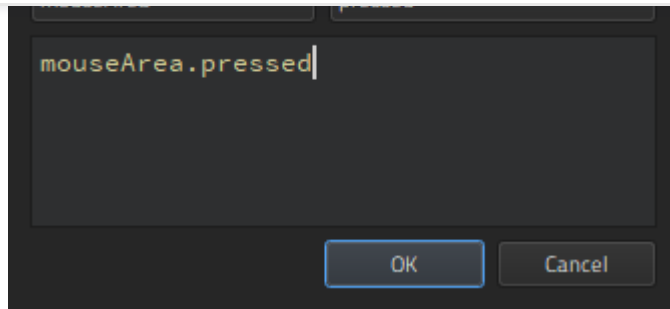
2. 选择“资源”以将要使用的图像文件复制到项目文件夹。
3. 在“导航”中，选择根组件，并在“属性”视图中设置按钮的宽度（W）和高度（H），以匹配您计划使用的图像的大小。这指定按钮组件的初始大小。
4. 将两个边框图像组件从“默认组件”>>“基本”拖放到“导航”中的根组件。
5. 将文本组件拖放到根组件。
6. 将鼠标区域拖放到根组件。
7. 选择边框图像以编辑其属性的值：
  - A. 在“Id”字段中，输入边框图像的ID。在此示例中，我们使用ID *非活动按钮*。
  - B. 在“源”字段中，选择边框图像的图像文件。例如，inactive\_button.png。
  - C. 在“布局”选项卡中，选择“（填充到父级）”按钮以始终使图像与其父级图像的大小相同。这使得按钮组件具有可伸缩性，因为图像大小绑定到组件大小。
8. 选择其他边框图像以类似方式编辑其属性的值：
  - A. 在“Id”字段中，输入“活动按钮”。
  - B. 在“源”字段中，单击按钮时为其选择图像文件。例如，active\_button.png。
  - C. 在“布局”选项卡中，选择“（填充到父项）”按钮。
9. 选择文本组件以在“属性”中指定字体大小和颜色：
  - A. 在“颜色”字段中，使用颜色选取器选择字体颜色，或在字段中输入一个值。
  - B. 在“字体”组的“大小”字段中，输入字体大小。
  - C. 在“布局”选项卡中，选择（垂直居中）和（水平居中）按钮以继承父项的垂直和水平居中。这可确保在调整组件大小时按钮标签居中。

## Using States to Change Component Property Values

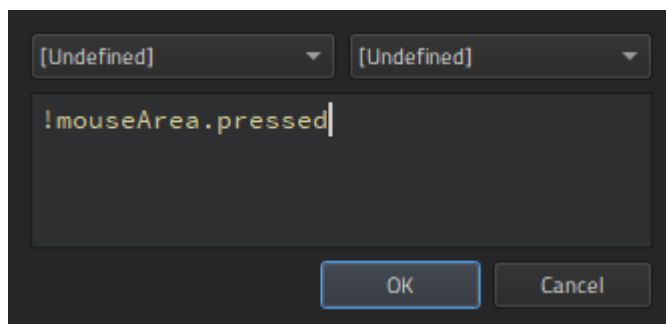
1. In the **States** view, select  twice to create two new states.




2. Select **State1**.
3. Change the state name to *active*.
4. Select , and then select **Set when Condition** to determine when the state should be applied.
5. In the **Binding Editor**, select the component and the signal to specify that the state is applied when the mouse button is pressed down.`mouseAreaPressed`



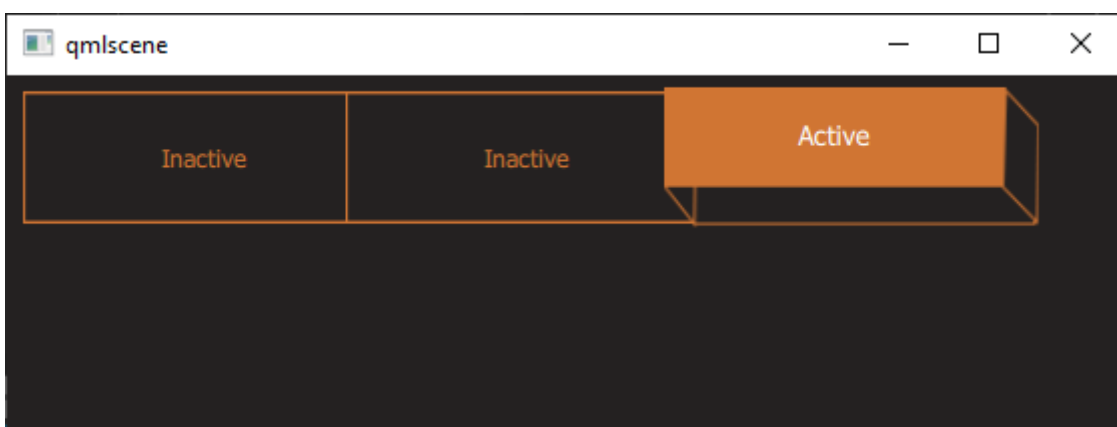
6. Select the text component in **Navigator** to specify that the text size is scaled up when the button is pressed down.
7. In **Properties**, select the **Advanced** section, and increase the value of the **Scale** property.
8. Select *inactiveButton* in **Navigator** to hide it in the *active* state by changing the value of its **Visibility** property in **Properties**.
9. Select **State2**.
10. Change the state name to *inactive*.
11. Set the when condition for the state to specify that the state is applied when the mouse button is not pressed down. `!mouseArea.pressed`



12. Press **Ctrl+S** to save the button.
13. Select the  (**Show Live Preview**) button to check how the button behaves when you click it. You can drag the preview window borders to see what happens when you resize the component.

To be useful, the button component has to be created in a project. When you work on other files in the project to create screens or other components for the UI, the button component appears in **Components > My Components**. You can drag-and-drop it to the **2D** or **Navigator** view to create button instances and modify the values of their properties to assign them useful IDs, change their appearance, and set the button text for each button instance, for example.

For more information about positioning buttons on screens, see [Scalable Layouts](#).



< [Creating Buttons](#)

[Specifying Component Properties](#) >



Contact Us

Company

- About Us
- Investors
- Newsroom
- Careers
- Office Locations

Licensing

- Terms & Conditions
- Open Source
- FAQ

Support

- Support Services
- Professional Services
- Partners
- Training

For Customers

- Support Center
- Downloads
- Qt Login
- Contact Us
- Customer Success

Community

- Contribute to Qt
- Forum
- Wiki
- Downloads
- Marketplace

