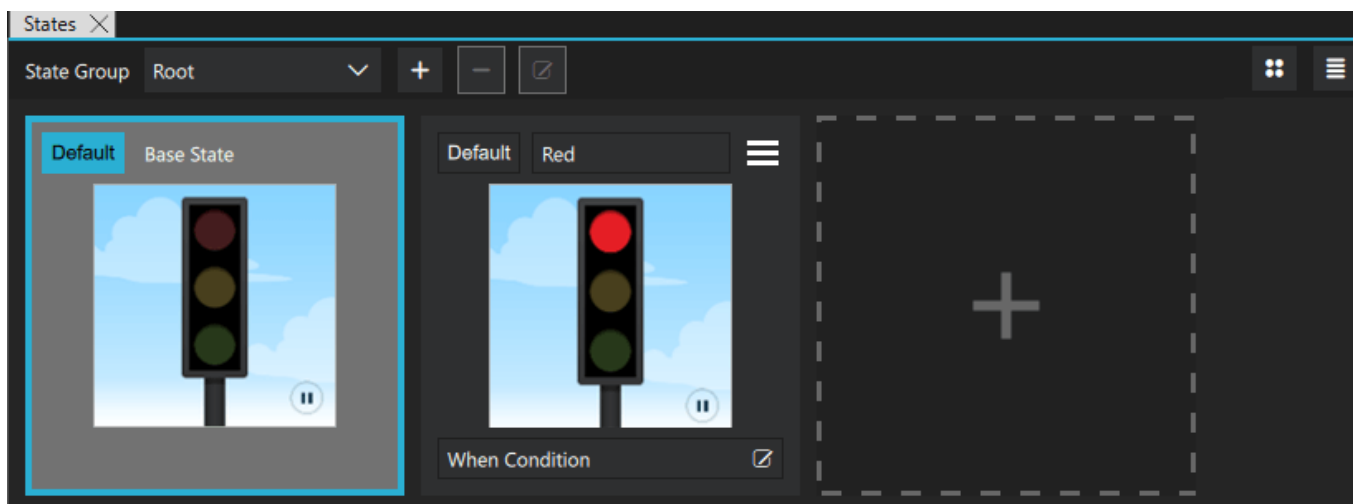


Qt设计工作室手册 > [与各国合作](#)

与各国合作

您可以通过选择在“[状态](#)”视图中定义组件和组件实例的状态⁺。

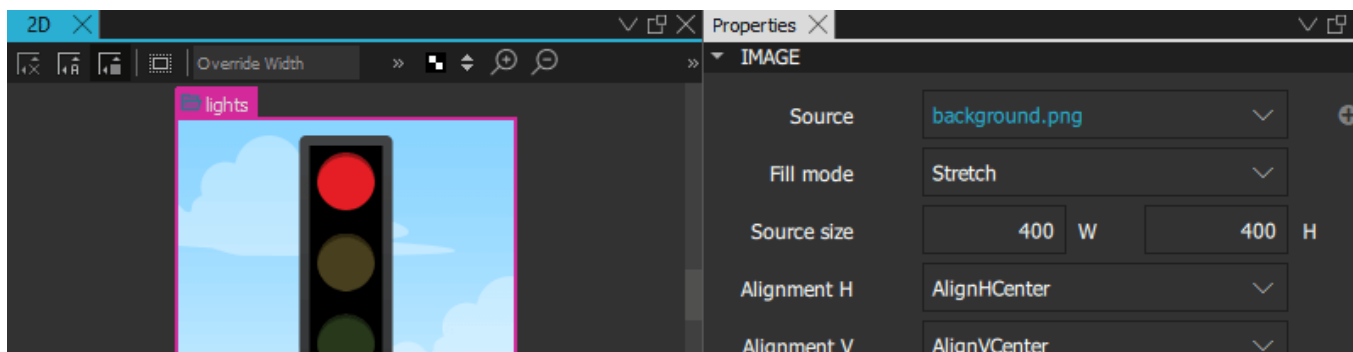


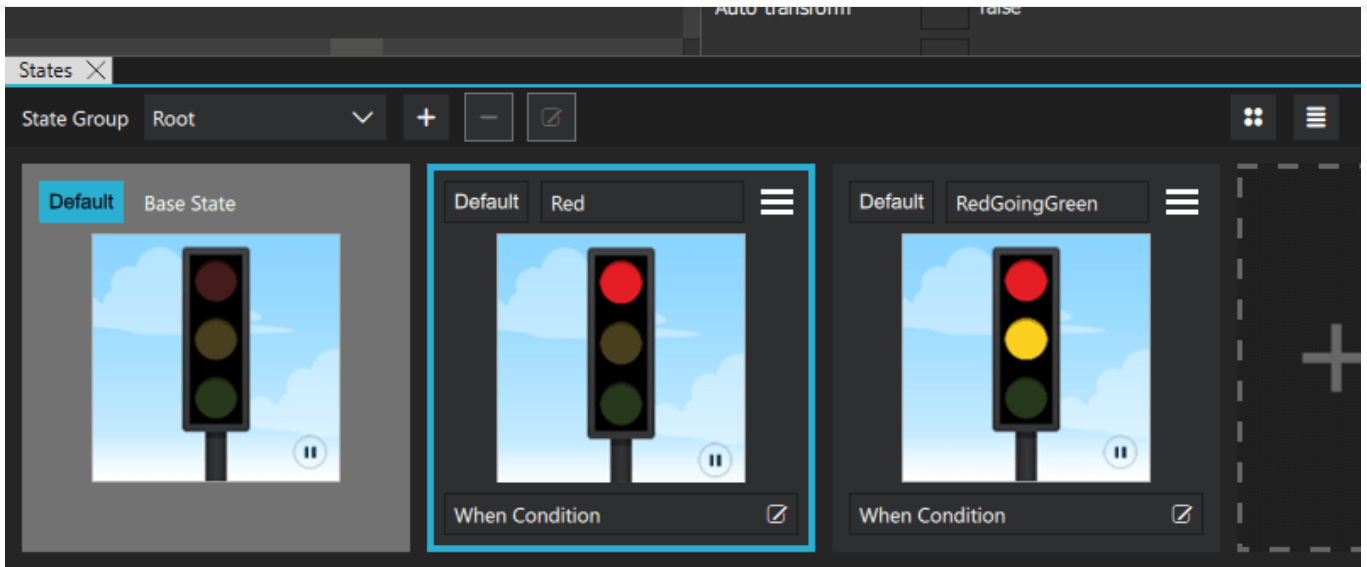
单击新状态以在2D和3D视图中切换到该状态，然后在“[属性](#)”中修改组件或组件实例的属性值。

例如，若要更改按钮的外观，可以在按钮组件中定义状态，以隐藏按钮图像并在其位置显示另一个图像，或者更改按钮背景或文本颜色。在其他组件中使用按钮实例时，可以通过隐藏或显示按钮组件实例来定义状态以创建不同的屏幕。**组件**>Qt快速控件>控件中的预设[按钮](#)控件具有预定义的[正常](#)和[关闭](#)状态。

这也适用于可以使用[向导模板](#)创建的自定义按钮组件。有关编辑按钮组件中的状态以及隐藏和显示按钮以创建多个屏幕的详细信息，请参阅[登录 UI - 组件](#)和[登录 UI - 状态](#)。要向屏幕添加运动，可以更改组件实例在2D视图中的位置，然后将动画添加到状态之间的更改中。

在某个状态下更改的属性将以蓝色突出显示。在“[代码](#)”视图中，可以看到记录为对基本状态的更改的更改。





注意：如果已在“**导航器**”中**锁定了组件**，并且尝试删除更改其属性值的状态，则系统会提示您确认删除。

有关更多信息，请观看以下视频：



设置默认状态

默认状态确定应用程序的启动状态。

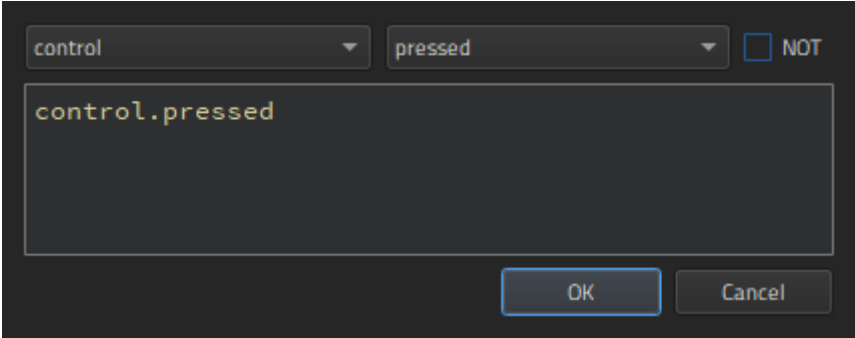
若要将状态设置为默认状态，请选择“**默认**”。

适用状态

若要确定何时应用状态，请在“**时间条件**”字段中选择 。在**绑定编辑器**中，指定状态的`when`属性。将属性的值设置为布尔表达式，该表达式的计算结果为何时要应用状态。true

这使您能够评估多个组件属性的真实性，并将 UI 移动到应用这些条件的状态。您可以评估某物是真还是假、大于或等于其他某物，等等。您还可以使用 AND 或 OR 等运算符来评估多个组件的真实性。

从左到右计算 `when` 条件，并按代码中出现的顺序计算。因此，如果两个不同状态的两个条件都计算为，则应用第一个状态。true



在**绑定编辑器**中编写表达式时，**代码完成**功能会列出可在表达式中使用的组件及其属性。

逻辑运算符摘要

可以在表达式中使用以下**逻辑运算符**在一个表达式中组合多个条件：

算子	意义	评估托伊夫true
!	不	不满足条件。
&&	和	这两个条件都满足。
	或	满足任一条件。
<	小于	左操作数小于右操作数。
>	大于	左操作数大于右操作数。
>=	大于或等于	左操作数大于或等于右操作数。
<=	小于或等于	左操作数小于或等于右操作数。
==	平等	操作数相等。
===	严格平等	操作数相等且类型相同。
!=	不等于	操作数不相等。
!==	严格不等于	操作数的类型相同但不相等，或者属于不同的类型。

或者，可以使用 And 运算符、Or 运算符和Not运算符组件，使用布尔运算符 AND、OR 和 NOT**运算符**绑定属性值。有关详细信息，请参阅**逻辑帮助程序**。

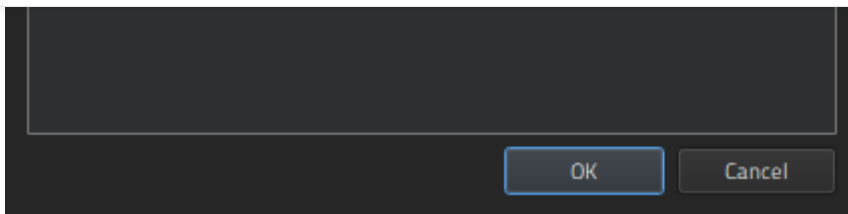
此外，您还可以使用算术运算符在检查前比较数字。但是，我们建议您尽可能为此目的创建单独的属性。

条件示例

要在按下按钮时将状态应用于按钮，只需编写：

```
when: control.pressed
```

若要在未按下按钮时应用状态，请选中“NOT”复选框。



若要在按钮未按下、选中或悬停时应用状态，可以组合条件，如下所示：

```
when: !control.pressed && !control.checked && !control.hovered
```

要在按下或选择按钮但未悬停时应用状态，您可以编写：

```
when: control.pressed || control.checked && !control.hovered
```

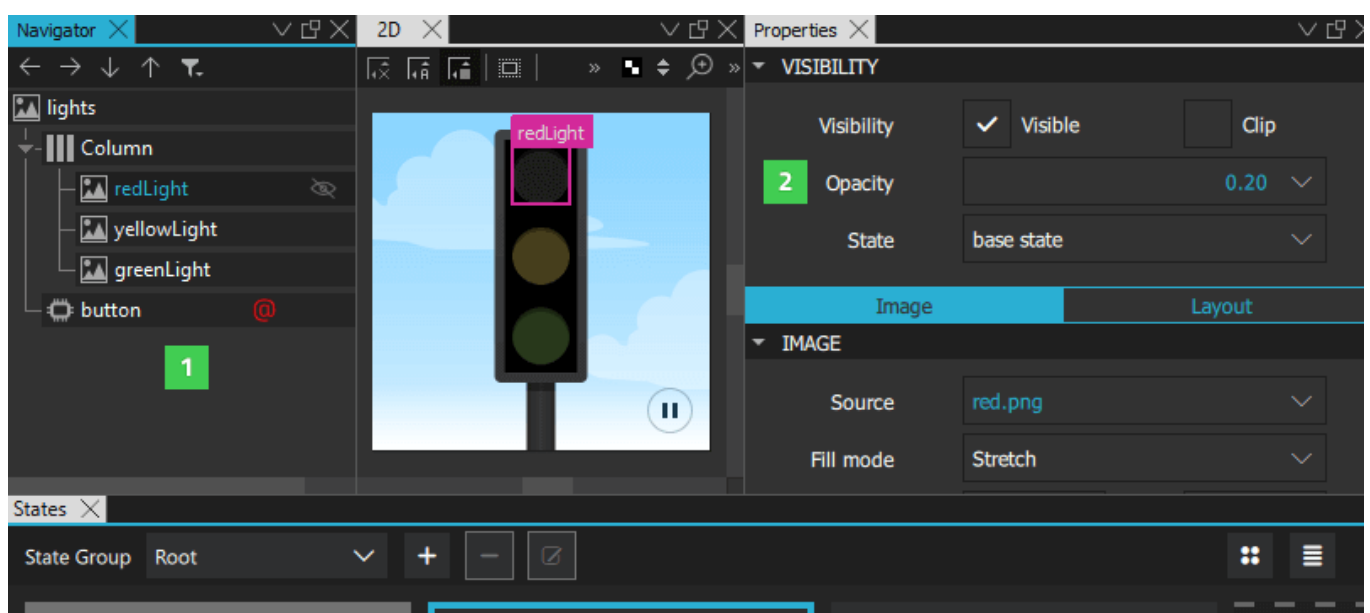
Qt设计工作室手册3.8.0
Topics >

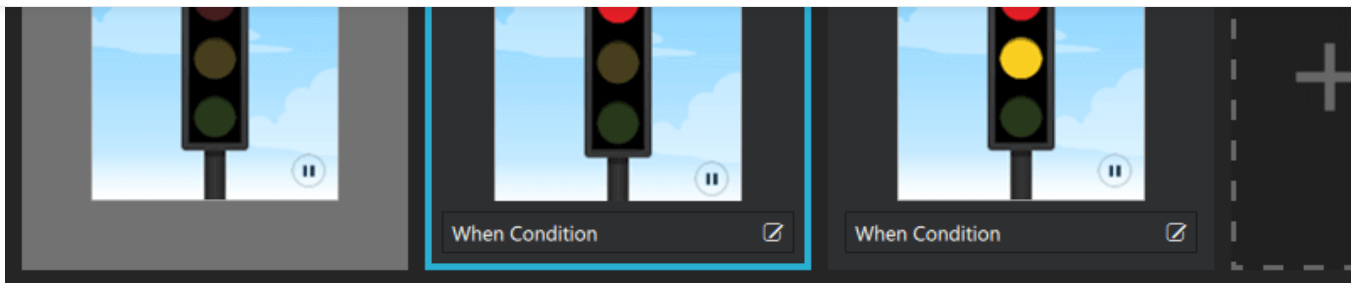
使用状态


若要保持代码干净，应创建一个基状态，其中包含应用程序所需的所有组件。然后，您可以创建状态，在其中隐藏和显示一组组件并修改其属性。这使您可以：

- 将不同视图上的组件彼此对齐。
- 避免过多的属性更改。如果组件在基状态下不可见，则必须将其子组件的所有更改定义为属性更改，这会导致代码复杂。
- 尽量减少基状态与其他状态之间的差异，以保持代码简短且可读，并提高性能。
- 在更改状态时使用过渡和动画时避免出现问题。

使用状态为应用程序创建视图：





1. 在基本状态下，添加应用程序中所需的所有组件（1）。处理一个视图时，可以单击“**导航**”中的图标以隐藏画布上不属于视图的组件。
2. 在“**状态**”中，选择 + 符号以创建新状态并为其命名。例如。Normal
3. 在“属性（2）”中，取消选中“**可见性**”复选框，或为此视图中不需要的每个组件设置“**不透明度**”为 0。如果指定父组件的设置，则所有子组件都将继承该设置，并且也会隐藏该设置。
4. 为每个视图创建其他状态，并设置视图中组件的可见性或不透明度。
5. 若要确定应用程序启动时应用的状态，请选择“**默认**”。

状态组

使用状态组，您可以独立于同一视图中的其他组件及其状态更改某些组件的状态。

每个状态组都有自己的属性更改和转换。

默认情况下，有一个根状态组。

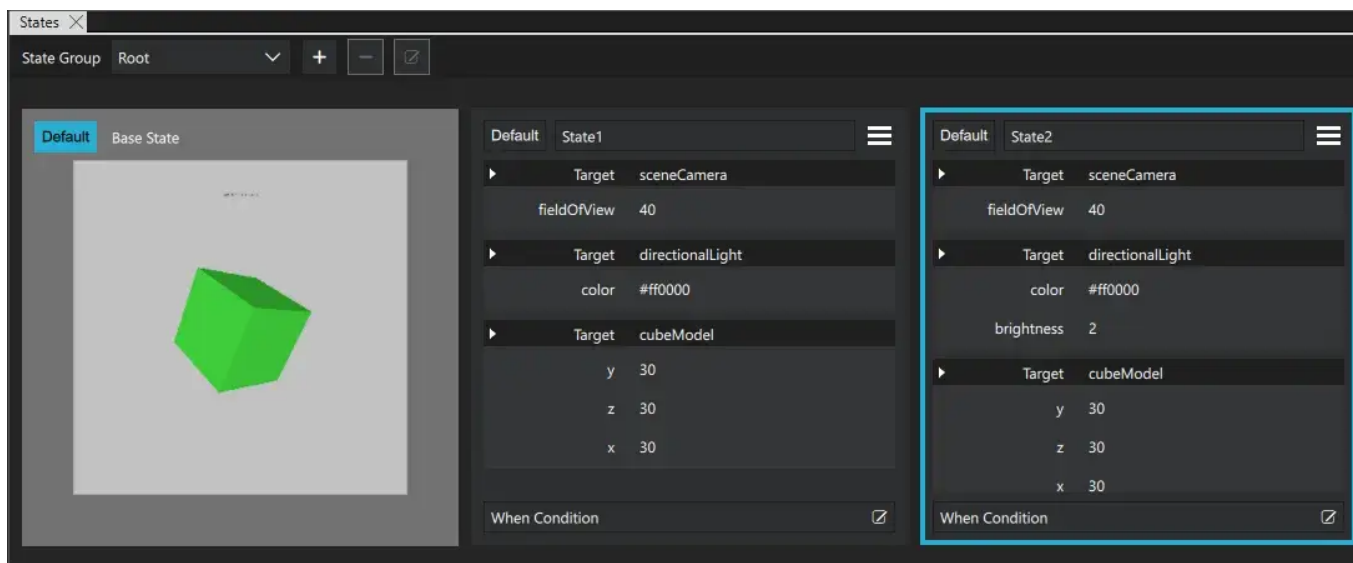
扩展状态

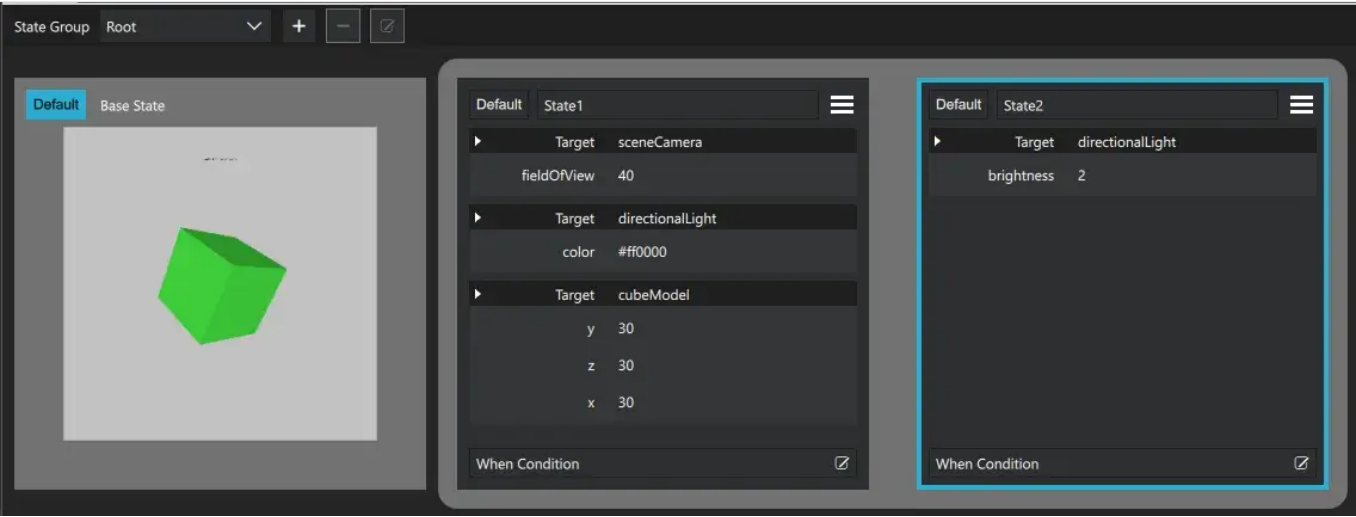
当一个状态扩展另一个状态时，它会继承该状态的所有更改。对于扩展状态指定的更改，正在扩展的状态被视为基本状态。

使用扩展状态可以使用户界面和 QML 代码更简洁。

下面是一个示例，其中状态 1 和状态 2 之间的唯一变化是方向光的亮度。

此处不使用扩展状态：





◀ 指定自定义属性

使用目标硬件进行验证 ▶



联系我们

公司

- 关于我们
- 投资者
- 编辑部
- 职业
- 办公地点

支持

- 支持服务
- 专业服务
- 合作 伙伴
- 训练

发牌

- 条款和条件
- 开源
- 常见问题

对于客户

- 支持中心
- 下载
- Qt登录
- 联系我们
- 客户成功案例

论坛
维基
下载
市场

© 2022 Qt公司

[反馈](#) [登录](#)