

🔍 搜索

Qt设计工作室手册 > [概念和术语](#)

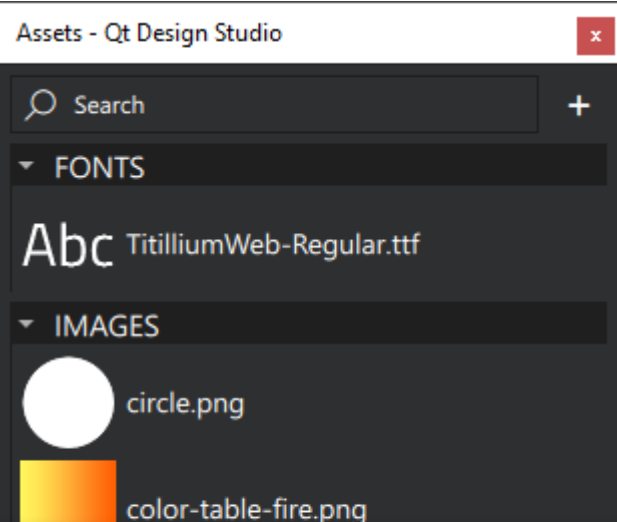
# 概念和术语

本主题介绍Qt Design Studio的主要概念和术语：

- › 资产
- › 捆绑
- › 元件
- › 连接
- › 装置
- › 模式
- › 项目
- › 财产
- › 信号
- › 州
- › 过渡

## 资产

资源是添加到项目的图像、字体文件、3D 模型或其他受支持的文件。



资产与组件一起打包，以便交付给用户。

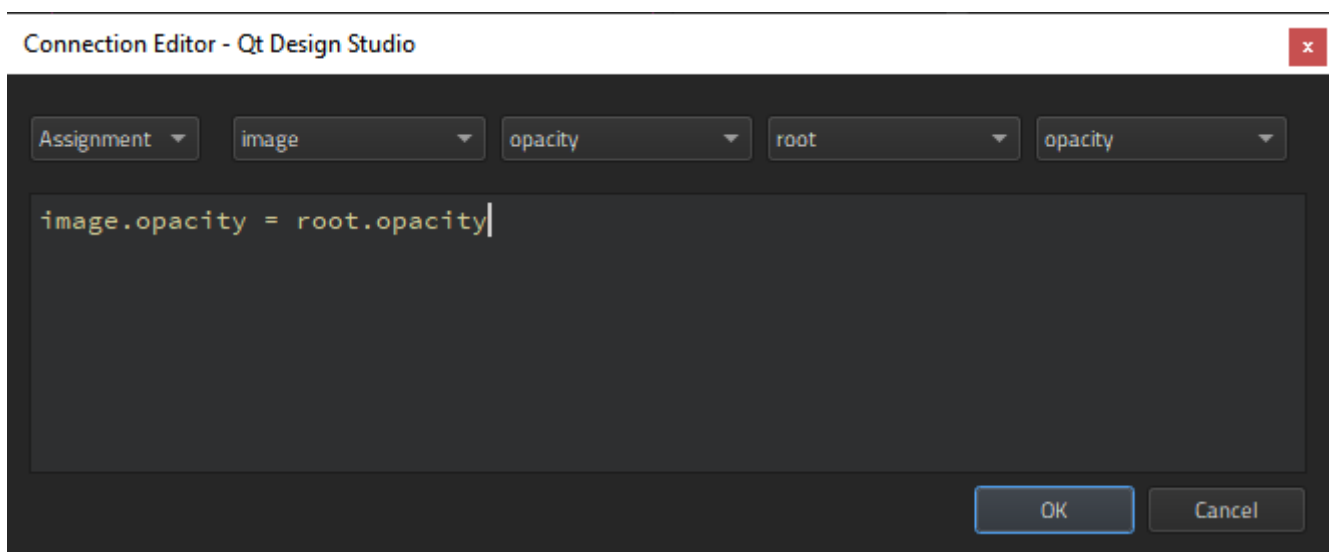
阅读有关资产的更多信息：

› 资产

## 捆绑

绑定是指定属性值的声明性方法。绑定允许将属性值表示为 JavaScript 表达式，该表达式定义相对于应用程序中可访问的其他属性值或数据的值。如果其他属性或数据值发生更改，则属性值将自动更新。

简单来说，绑定可能是对另一个属性的引用。例如，组件的高度可以绑定到其父组件的高度，以便在父组件高度更改时，会自动调整组件高度。同样，组件的不透明度可以绑定到其父组件的不透明度。



每当为属性分配 JavaScript 表达式时，都会隐式创建属性绑定。

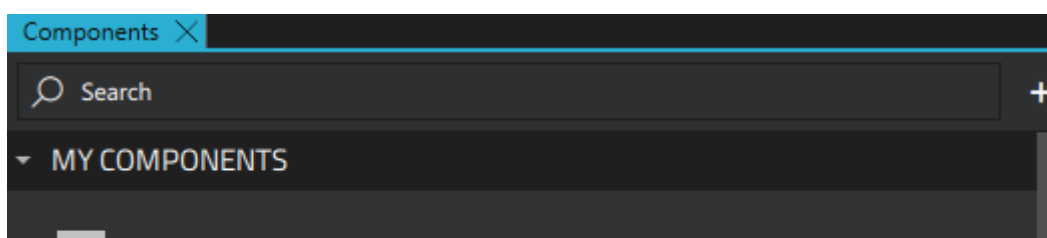
阅读有关绑定的更多信息：

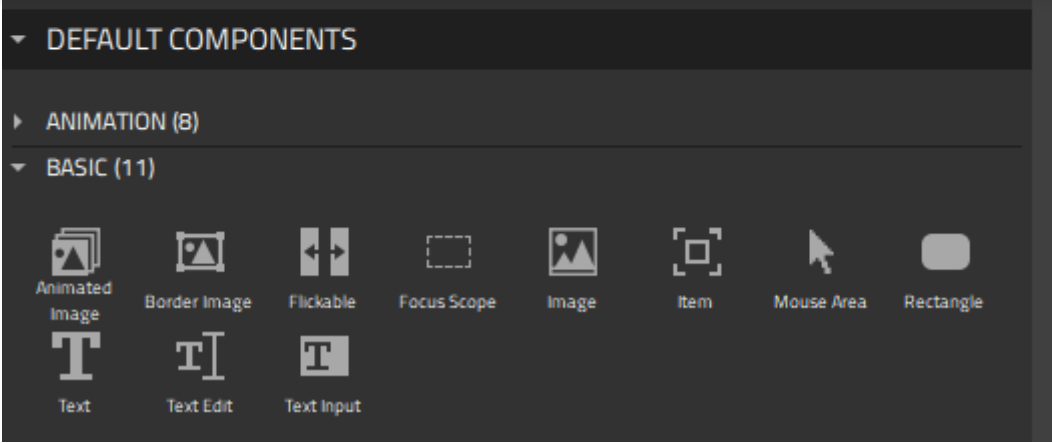
- › 在属性之间添加绑定
- › 设置绑定

## 元件

组件是 UI 的可重用构建基块。

Qt Design Studio 附带了预设组件，您可以通过创建它们的实例在 UI 中使用它们。这些类似于草图中的符号或 Unity 中的预制件。





一些预设组件表示简单的形状、文本或图像，而其他组件表示具有完整功能的复杂 UI 控件，如数字调整框或滑块。您还可以将预设3D 组件的实例添加到 UI。您可以在组件中找到所有预设组件。

要构建自己的组件，您可以修改组件实例的属性并组合它们。

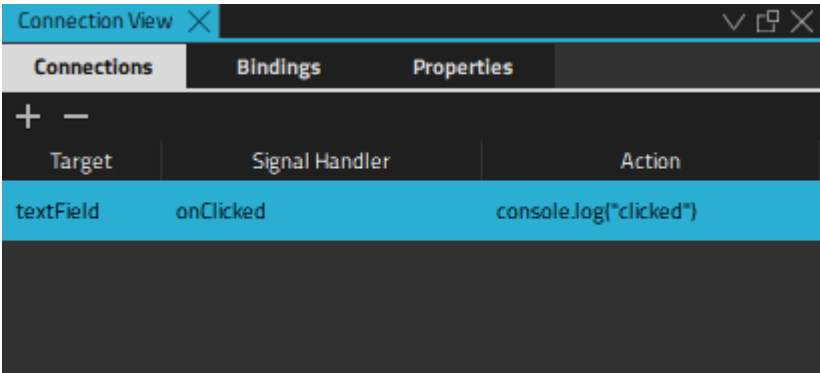
组件在一个文件中指定（文件扩展名为`ui.qml`或`.qml`）。例如，可以在 `Button.ui.qml` 中定义按钮组件。通常，组件的可视外观是在 UI 文件中定义的。若要创建组件文件，可以使用向导模板，或将组件实例移动到单独的组件文件中。

阅读有关组件的更多信息：

- › 预设组件
- › 创建组件实例
- › 创建自定义组件

## 连接

可以在组件和信号之间创建连接，以确定 UI 应如何响应应用程序事件。在组件之间创建连接的另一种方法是在其属性值之间创建绑定。



阅读有关连接的更多信息：

- › 连接
- › 使用连接

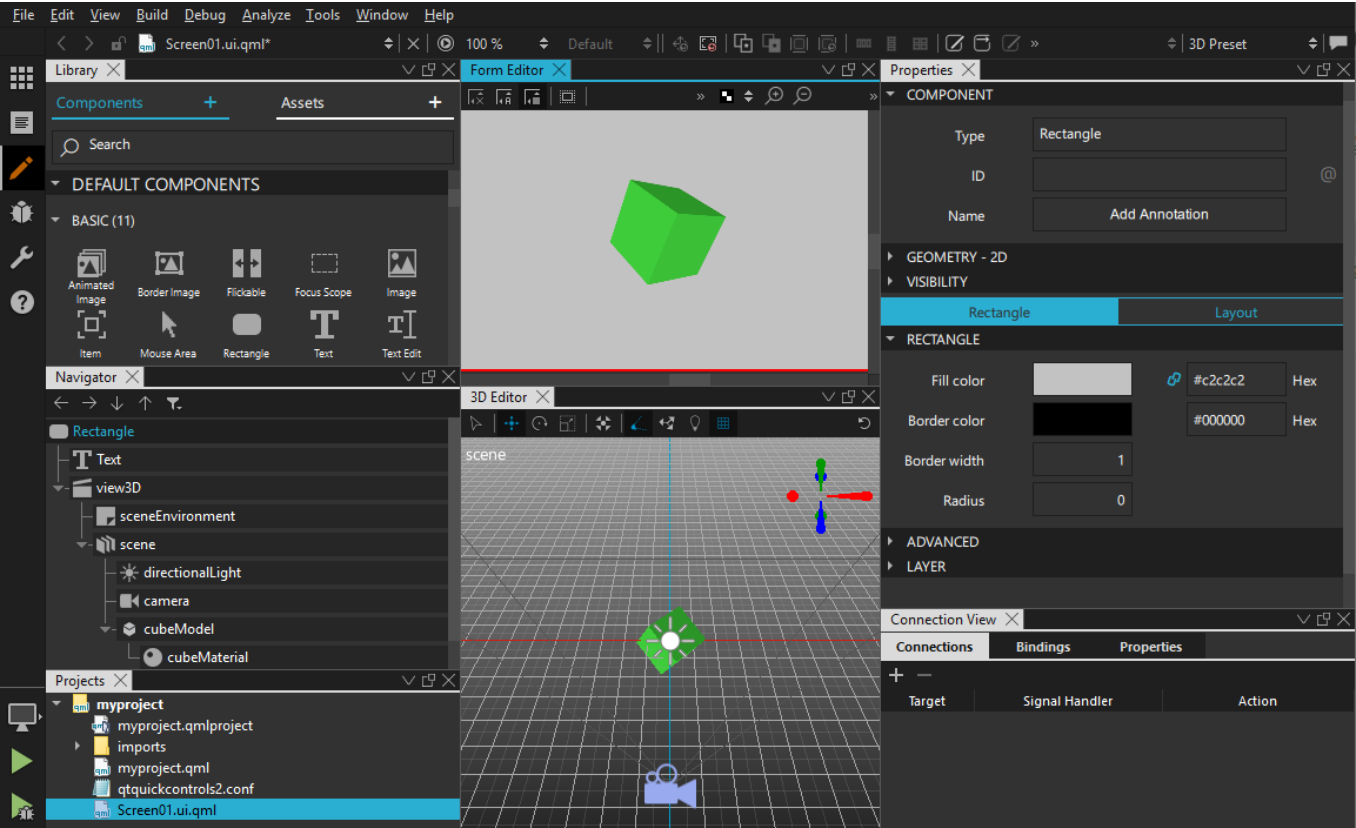
## 装置

嵌入式设备。

## 模式

模式使Qt Design Studio UI适应手头的不同UI设计任务。每种模式都有自己的视图，该视图仅显示执行特定任务所需的信息，并仅提供与之相关的最相关的特性和功能。因此，Qt Design Studio窗口的大部分区域始终专用于实际任务。

对于设计师来说，最重要的模式是针对实际工作的**设计**、**欢迎**打开示例和教程以及帮助阅读文档。其他模式是应用程序开发最需要的。



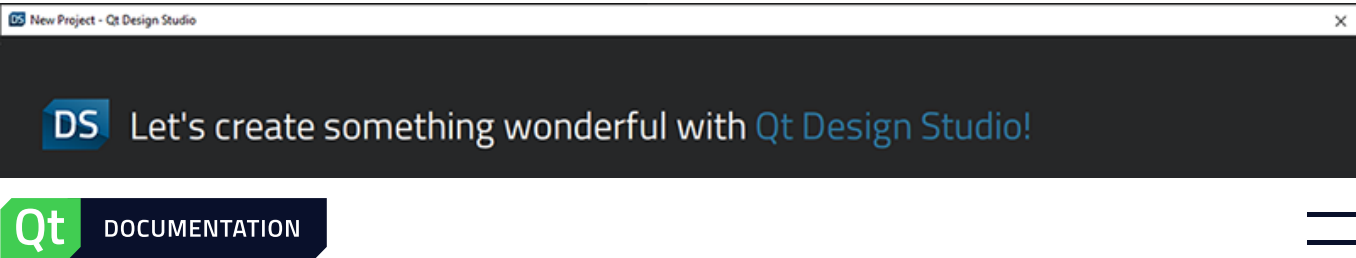
阅读有关模式的更多信息：

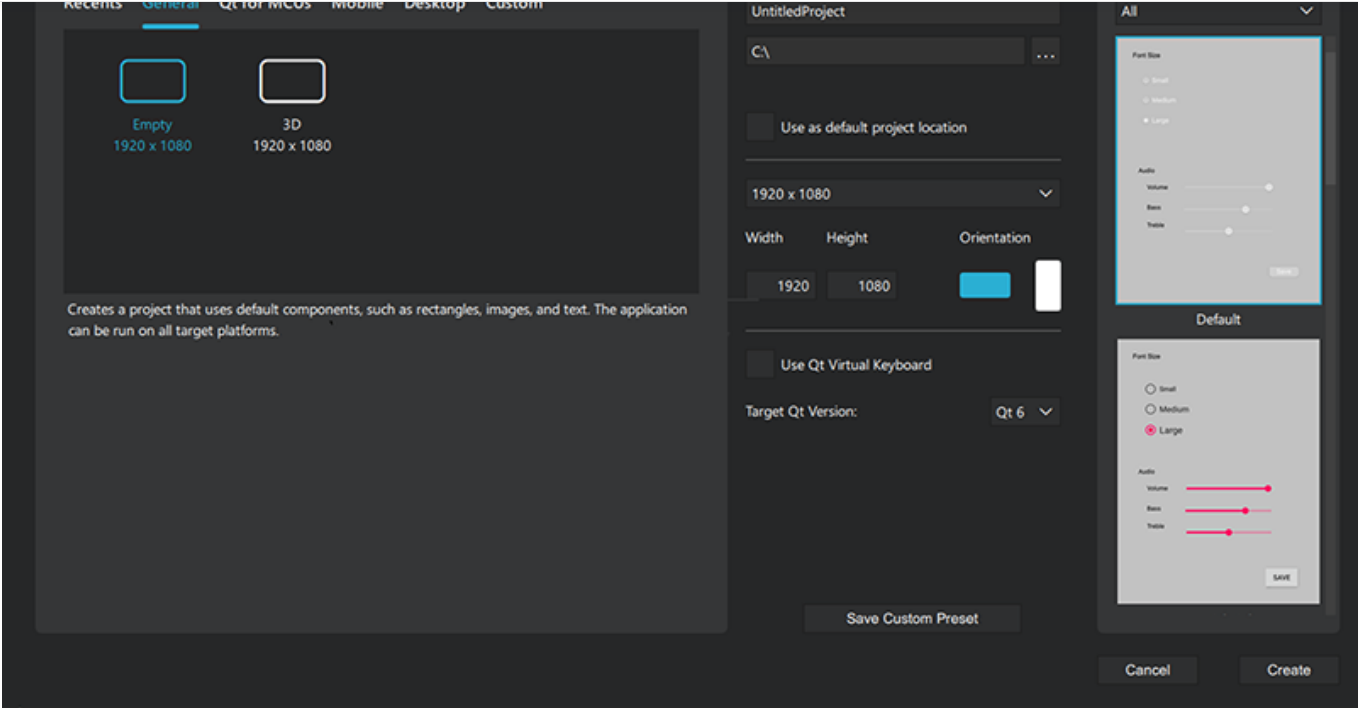
- 选择模式
- 设计视图

## 项目

Qt设计工作室手册3.8.0  
Topics >

您可以根据需要使用模板创建不同类型的项目。默认情况下，模板会将预设组件添加到项目中。例如，如果创建 3D 项目，则会向其添加预设的 3D 组件。您可以在组件中添加更多预设**组件**。



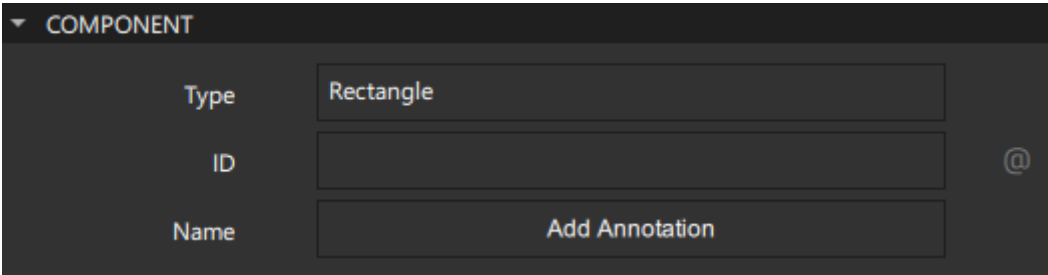


阅读有关项目的更多信息：

› [创建项目](#)

## 财产

属性是组件的属性，可以为其分配静态值或绑定到动态表达式。属性的值可以由其他组件读取。通常，它也可以由另一个组件修改，除非特定组件类型明确禁止对特定属性执行此操作。



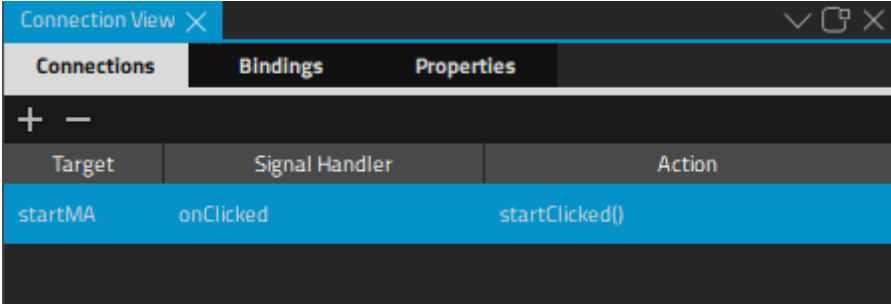
阅读有关属性的更多信息：

- › [性能](#)
- › [预设组件](#)
- › [指定组件属性](#)
- › [在属性之间添加绑定](#)
- › [指定自定义属性](#)

## 信号

信号表示应用程序事件，例如用户单击按钮或组件的属性值更改。应用程序需要此信息来执行操作或将其中继到其他应用程序。

信号是Qt框架的一部分，用于在应用程序的不同部分之间传递信息。信号可以连接到槽函数，当信号发出时，槽函数将执行指定的操作。信号和槽是Qt框架的核心概念之一，用于实现事件驱动编程。



此外，当属性的值发生变化时，会自动发出信号。

阅读更多关于信号的信息：

- › 将组件连接到信号
- › 鼠标区域

## 州

特定可视组件的状态是一组信息，用于描述组件的各个部分在其中的显示方式和位置，以及与该状态关联的所有数据。UI 中的大多数可视组件将具有有限数量的状态，每个状态都具有明确定义的属性。

例如，列表中的某个元素可能被选中，也可能未被选中，如果选定，则该元素可能是当前活动的单个选择，也可能是选择组的一部分。其中每个状态可能具有某些关联的视觉外观（中性突出显示、展开等）。

同样，按钮的外观可以更改以指示按下状态。



阅读有关状态的更多信息：

- › 国家
- › 与各国合作

## 过渡

当可视组件从一种状态转换到另一种状态时，其外观会发生变化。过渡是两个状态之间的边。它可能会触发其他事件的发生，因为应用程序的其他部分可能具有在进入或离开特定状态时触发的行为。

阅读有关过渡的更多信息：

- › 转换
- › 为状态之间的过渡添加动画效果

‹ 使用案例

例子 ›



The Qt  
Company



联系我们

公司

关于我们

投资者

编辑部

职业

办公地点

发牌

条款和条件

开源

常见问题

支持

支持服务

专业服务

合作伙伴

训练

对于客户

支持中心

下载

Qt登录

联系我们

客户成功案例

社区

为Qt做贡献

论坛

维基

下载

市场

© 2022 Qt公司

反馈 登录