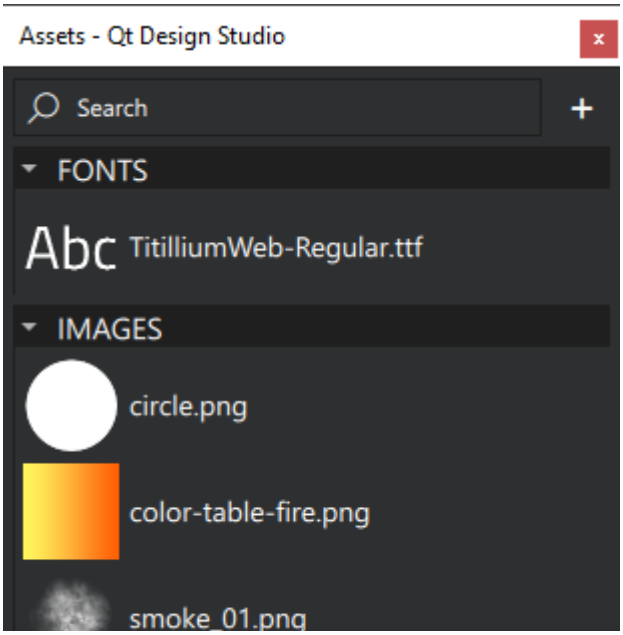
概念和术语

本主题介绍Qt设计工作室的主要概念和术语：

资产

*资源*是添加到项目中的图像、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。

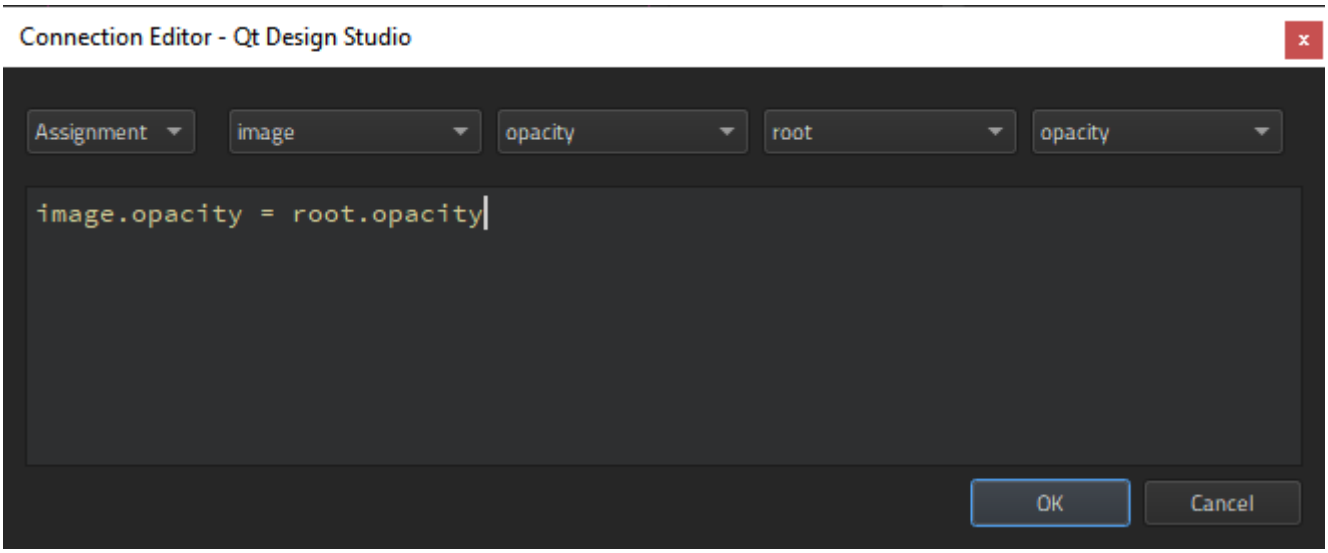
阅读有关资产的更多信息：

- 资产

捆绑

*绑定*是指定属性值的一种声明性方法。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，则属性值会自动更新。

在最简单的情况下，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级的高度，以便当父级高度发生变化时，自动调整零部件高度。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。

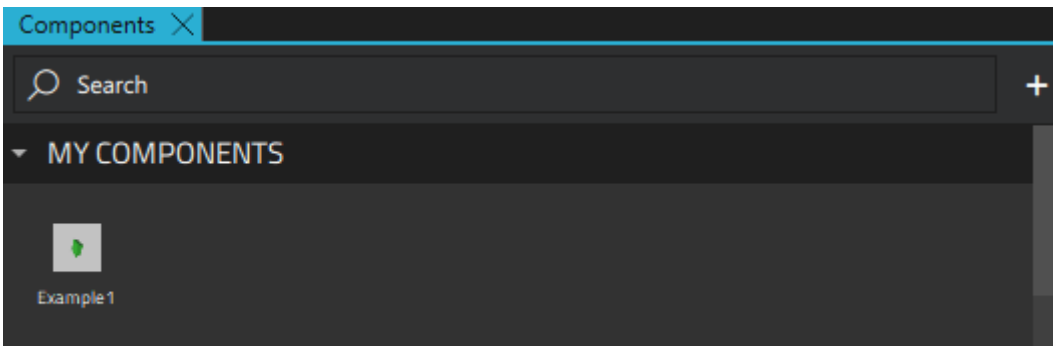
阅读有关绑定的更多信息：

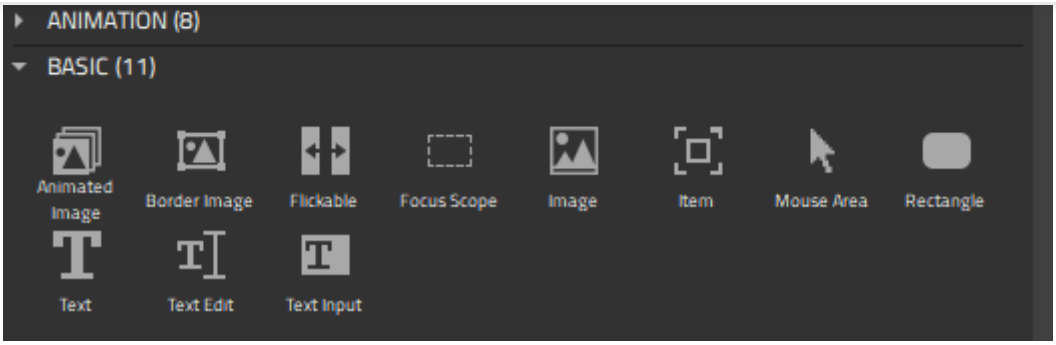
- 在属性之间添加绑定
- Setting Bindings

Component

A *component* is a reusable building block for a UI.

Qt Design Studio comes with *preset components* that you can use in your UI by creating instances of them. These are similar to *Symbols* in Sketch or *Prefab* in Unity.





Some of the preset components represent simple shapes, text, or images, while others represent complex UI controls with full functionality, such as spin boxes or sliders. You can also add instances of preset 3D components to your UIs. You can find all the preset components in Components.

To build your own components, you can modify the properties of the component instances and combine them.

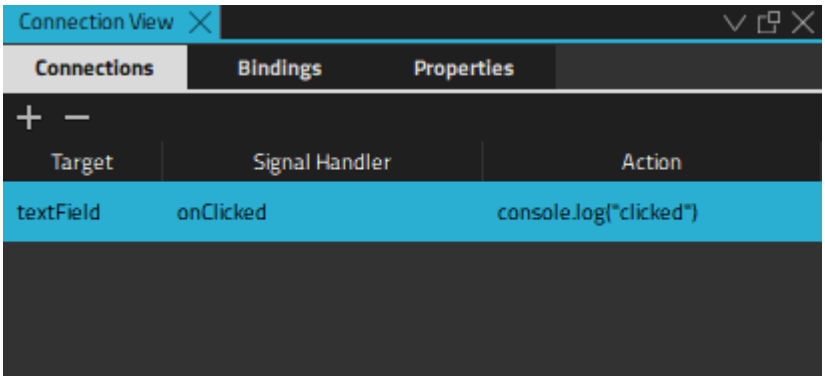
A component is specified within one file (with the file extension *ui.qml* or *.qml*). For example, a Button component may be defined in Button.ui.qml. Typically, the visual appearance of a component is defined in a *UI file*. To create component files, you can use wizard templates, or move component instances into separate component files.

Read more about components:

- Preset Components
- Creating Component Instances
- Creating Custom Components

Connection

A *connection* can be created between a component and signal to determine how the UI should react to application events. Another way to create connections between components is to create bindings between the values of their properties.



Read more about connections:

- Connections
- Adding Connections

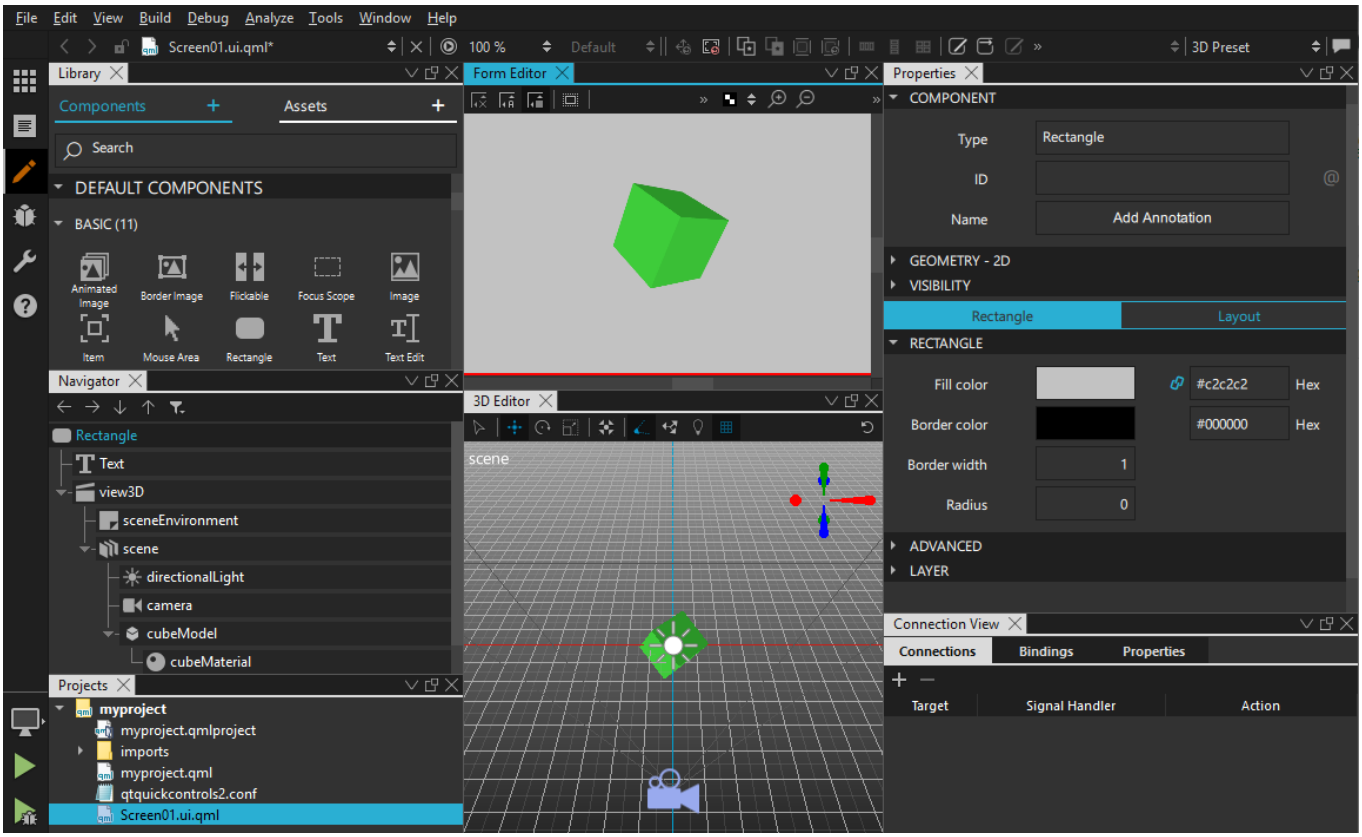
Device

An embedded device.

Mode

A *mode* adapts the Qt Design Studio UI to the different UI design tasks at hand. Each mode has its own view that shows only the information required for performing a particular task, and provides only the most relevant features and functions related to it. As a result, the majority of the Qt Design Studio window area is always dedicated to the actual task.

For a designer, the most important modes are **Design** for the actual work, **Welcome** for opening examples and tutorials, and **Help** for reading documentation. The other modes are mostly needed for application development.



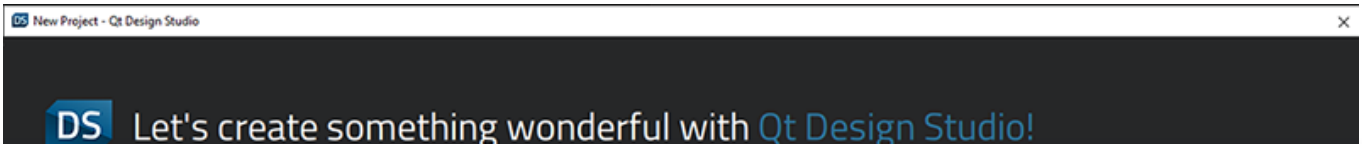
Read more about modes:

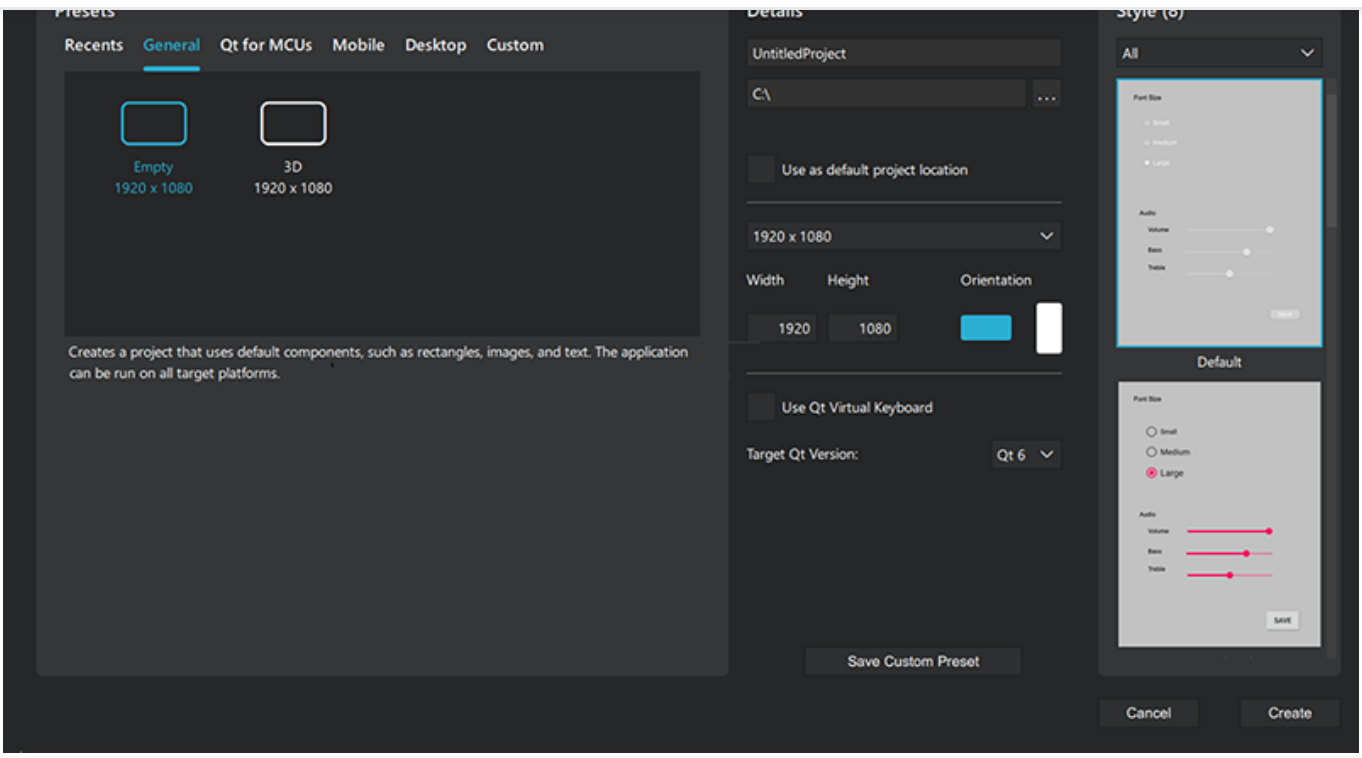
- Selecting Modes
- Design Views

Project

A project is a container for the components and assets that you use in your UI. You can *package* the UI and preview or run it on different operating systems on the desktop or a device.

You use templates to create different types of projects according to your needs. The templates add preset components to the project by default. For example, if you create a 3D project, preset 3D components are added to it. You can add more preset components in **Components**.



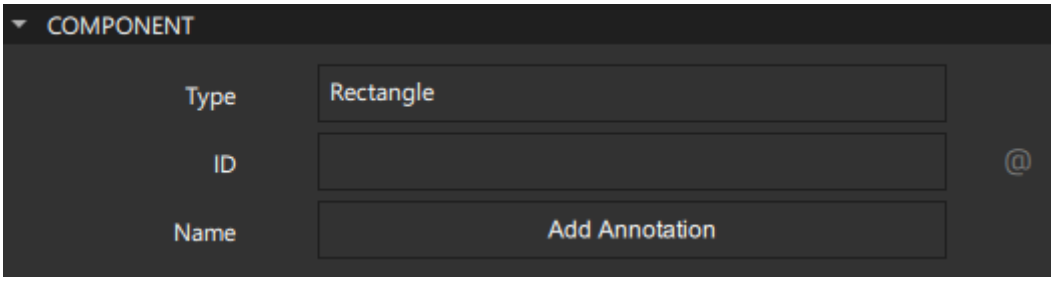


Read more about projects:

- Creating Projects

Property

A *property* is an attribute of a component that can be assigned a static value or bound to a dynamic expression. A property's value can be read by other components. Generally, it can also be modified by another component, unless a particular component type has explicitly disallowed this for a specific property.



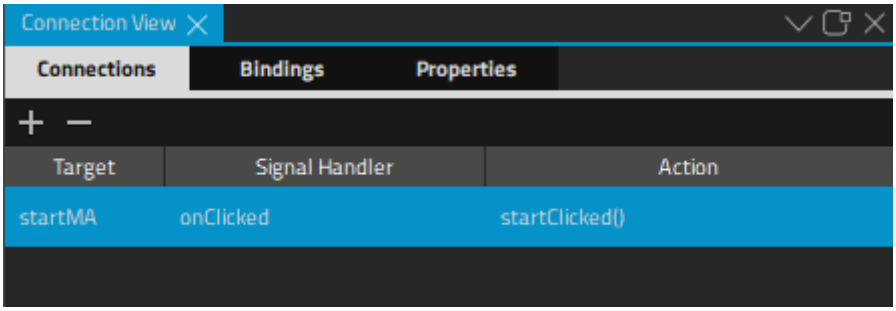
Read more about properties:

- Properties
- Preset Components
- Specifying Component Properties
- Adding Bindings Between Properties
- Specifying Custom Properties

Signal

A *signal* represents an application event, such as a user clicking a button or the value of a property of a component

Mouse Area component has a signal that is emitted whenever the mouse is clicked within the area. Since the signal name is , the signal handler for receiving this signal is named .`clicked``clickedonClicked`



Further, a signal is automatically emitted when the value of a property changes.

Read more about signals:

- Connecting Components to Signals
- Mouse Area

State

The *state* of a particular visual component is the set of information that describes how and where the individual parts of the component are displayed within it, and all the data associated with that state. Most visual components in a UI will have a limited number of states, each with well-defined properties.

For example, an element in a list may be either selected or not, and if selected, it may either be the currently active single selection or it may be part of a selection group. Each of those states may have certain associated visual appearance (neutral highlighted, expanded, and so forth).

Similarly, the appearance of a button can change to indicate a *pressed* state.

Read more about states:

- States
- Adding States

Transition

When a visual component transitions from one state to another, its appearance changes. A *transition* is an *edge* between two states. It may trigger other events to occur, as other parts of the application may have behavior that is triggered when a certain state is entered or left.

Read more about transitions:

- Transitions
- Animating Transitions Between States

‹ Use Cases Examples ›

Qt DOCUMENTATION

The Qt Company

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

Feedback Sign In