Qt **DOCUMENTATION**

Search   Topics ❯

Qt Design Studio Manual ❯ Specifying Custom Properties

Specifying Custom Properties

Each preset component has a set of preset properties that you can specify values for. You can add custom properties that would not otherwise exist for a particular component type. You bind the properties to dynamic expressions to define global properties for a component that can be read by other components. For example, you can specify global properties for the root component that you can use in the child components.

For example, to specify spacing between UI elements, you could define a margin for a component that does not have a margin property, and then use bindings to refer to the value of the margin property from other components.

Similarly, you can add custom properties for your own components that are based on preset components.

Any content that is data-driven should be exported as a public property (alias property) of the relevant component. For example, a speedometer should have an *int* or *real* property for speed to which the UI is bound.
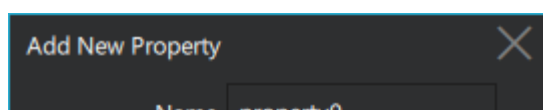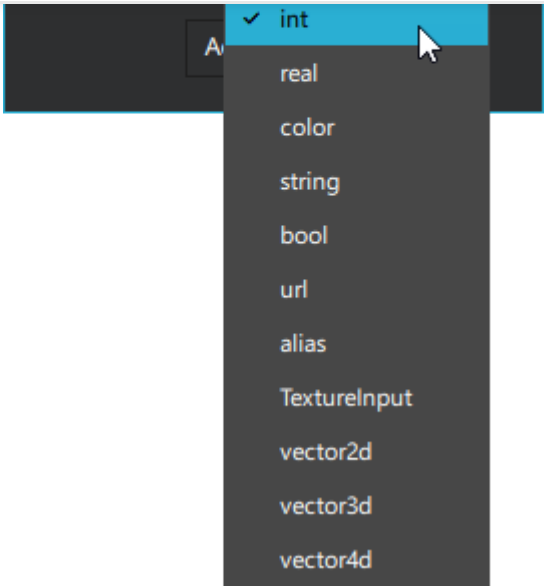
Adding Properties for a Component

To add a custom property for a component:

1. Go to the **Local Custom Properties** section in the **Properties** view.
2. Select the ➕ (**Add**) button to add a custom property for the currently selected component.
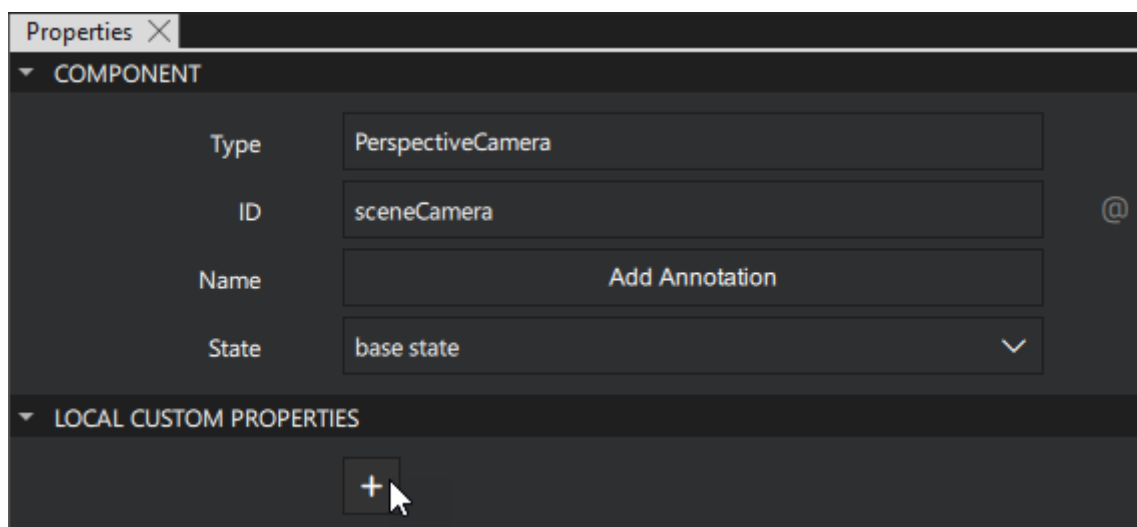
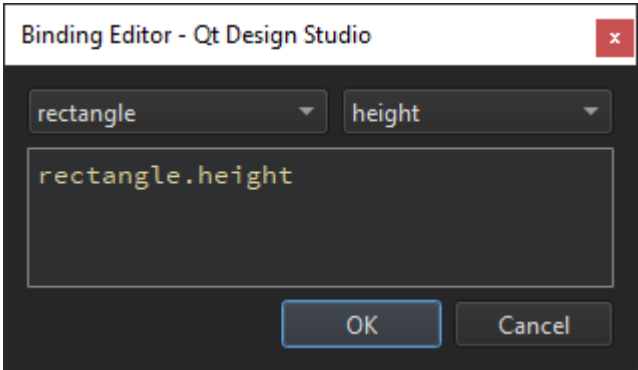3. Set the **Name** and **Type** for the property.

Binding a Property Value

To bind the value of the property to that of another one or to data accessible in the application.

1. In the **Properties** view, select ⬡ next to the property.
2. Select **Set Binding**.

For more information, see Setting Bindings.

Supported Property Types

The following table describes the supported property types:

Type	Description
alias	Property alias that holds a reference to another property
bool	Binary `true` or `false` value
color	Color value that can be specified by using an SVG color name, such as "red", "green", or "lightsteelblue", or a hexadecimal triplet or quad in the form "#RRGGBB" and "#AARRGGBB", respectively. For example, the color red corresponds to a triplet of "#FF0000" and a slightly transparent blue to a quad of "#800000FF". In addition, you can use the following Qt functions: Qt.rgba(), Qt.hsva(), Qt.hsla(), Qt.darker(), Qt.lighter(), and Qt.tint().
Type	Description

Qt DOCUMENTATION

string	Free form text string
TextureInput	Specifies a texture exposed to the shaders of a CustomMaterial or Effect.
url	Resource locator, such as a file name. It can be either absolute, (`http://qt-project.org`), or relative (`pics/logo.png`). A relative URL is resolved relative to the URL of the parent component.
variant	Generic property type. For example, variant properties can store numbers, strings, objects, arrays, and functions.
vector2d	Refers to a value with x and y attributes.
vector3d	Refers to a value with x, y, and z attributes.
vector4d	Refers to a value with x, y, z, and w attributes.

See also Specifying Component Properties.

< Adding Bindings Between Properties   Adding States >

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

Qt DOCUMENTATION

Contribute to Qt

Forum

Wiki

Downloads

Marketplace

Feedback   Sign In