

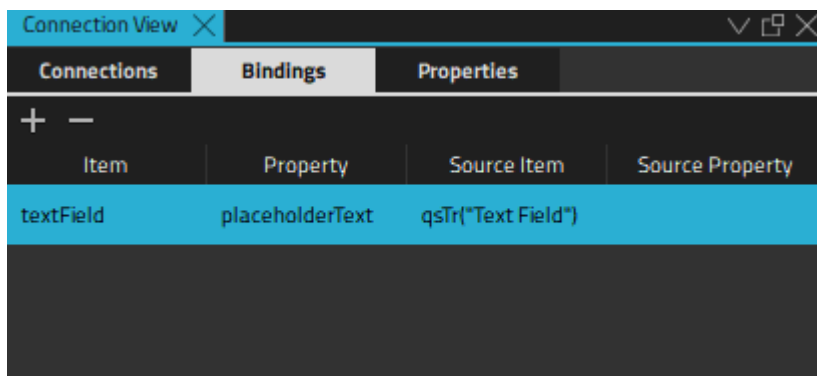
Adding Bindings Between Properties

To dynamically change the behavior of a component, you can create a *binding* between the properties of two components. To create a property binding, a property is assigned a JavaScript expression that evaluates to the desired value. Behind the scenes, the variables in the binding expression are monitored. When a change is detected, the binding expression is re-evaluated and the new result is applied to the property.

At its simplest, a binding may be a reference to another property. For example, the height of a component can be bound to the height of its parent, so that when the parent height changes, the component height is adjusted automatically.

For more information about the use of property bindings, see [Property Binding](#).

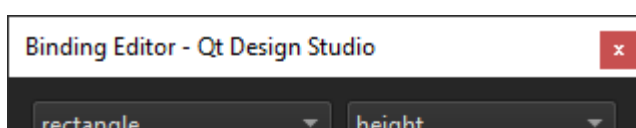
You can create bindings between components in **Bindings**.

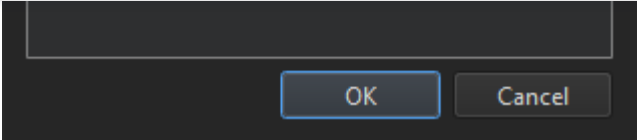


To bind a property of a component to the property of another component:

1. Go to the **Binding** tab in the [Connections](#) view.
2. Select the **+** (Add) button to add a binding for the currently selected component. The component ID is displayed in the **Item** column.
3. Double-click the value in the **Property** column to select the property to bind to a source property.
4. Double-click the value in the **Source Item** column to select the component whose property you want to use to determine the behavior of the target component.
5. Double-click the value in the **Source Property** column to select the property to bind the target property to.

Right-click a binding and select **Open Binding Editor** in the context menu to specify the binding as a JavaScript expression in **Binding Editor**. For more information, see [Setting Bindings](#).





For examples of creating property bindings, see:

- › [Using States to Change Component Property Values](#)
- › [Exporting Properties](#)

For more information, watch the following video:

Summary of Logical Operators

You can use the following **logical operators** in the expressions to combine several conditions in one expression:

Operator	Meaning	Evaluates to if true
!	NOT	The condition is not met.
&&	AND	Both conditions are met.
	OR	Either of the conditions is met.
<	Less than	The left operand is less than the right operand.
>	Greater than	The left operand is greater than the right operand.
>=	Greater than or equal	The left operand is greater than or equal to the right operand.
<=	Less than or equal	The left operand is less than or equal to the right operand.
==	Equal	The operands are equal.
===	Strict equal	The operands are equal and of the same type.
!=	Not equal	The operands are not equal.
!==	Strict not equal	The operands are of the same type but not equal, or are of different type.

Alternatively, you can use **And Operator**, **Or Operator**, and **Not Operator** components to bind property values using the boolean AND, OR, and NOT operator. For more information, see [Logic Helpers](#).

In addition, you can use arithmetic operators to compare numbers before checks. However, we recommend that



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

