Connecting Components to Signals

A signal and handler mechanism enables components to respond to application events, which are represented by *signals*. When a signal is emitted, the corresponding *signal handler* is invoked to respond to the event by using scripts or other operations placed in the handler.

To receive a notification when a particular signal is emitted for a particular component, the component definition should declare a signal handler named *on*<*Signal*> where <*Signal*> is the name of the signal, with the first letter capitalized. The signal handler should contain the JavaScript code to be executed when the signal handler is invoked.

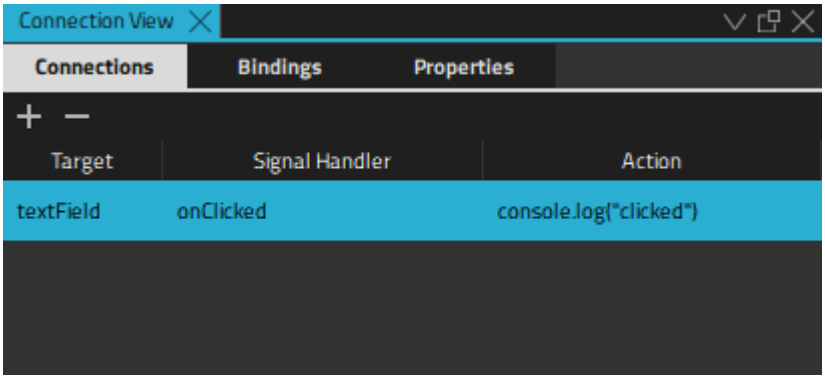
Components have predefined signals that are emitted when users interact with the application. For example, the Mouse Area component has a `clicked` signal that is emitted whenever the mouse is clicked within the area. Since the signal name is `clicked`, the signal handler for receiving this signal is named `onClicked`.

A signal is automatically emitted when the value of a property changes. This type of signal is a *property change signal* and signal handlers for these signals are written in the form `on<Property>Changed`, where <*Property*> is the name of the property, with the first letter capitalized.

For example, the Mouse Area component has a `pressed` property. To receive a notification whenever this property changes, you would use a signal handler called `onPressedChanged`.

For more information about signals and signal handlers, see Signal and Handler Event System.

You can connect components to signals that are available to them in **Connections**.



To connect components to signals:

1. Go to the **Connections** tab in the Connections view.
2. Select the + (**Add**) button to add a connection.
3. Double-click the value in the **Target** column to add the component to connect to a signal.
4. Double-click the value in the **Signal Handler** column to select the signal that the connection will listen to from a list of all signals available for the component.

Right-click a connection and select **Open Connection Editor** in the context menu to specify the connection in **Connection Editor**.

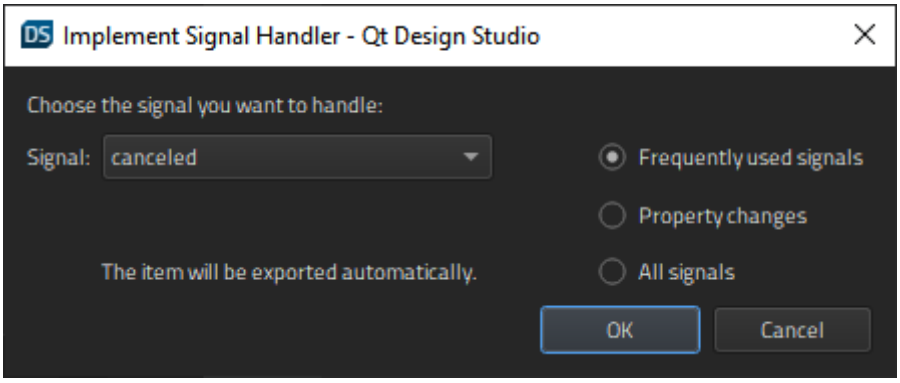
For examples of using the **Connections** view, see:

- Connecting Buttons to States in Log In UI - States
- Connecting Buttons to State Changes in Washing Machine UI

Adding Signal Handlers

If a signal handler that you need is not listed in the **Signal Handler** column, you can add it:

1. Right-click a component in the Navigator or 2D view and select **Add New Signal Handler** in the context menu.
2. In the **Signal** field, select the signal to handle.



3. Select the radio buttons to filter the list to only display frequently used signals or property changes.
4. Select **OK**.

The added signal handler is automatically exported as a property.

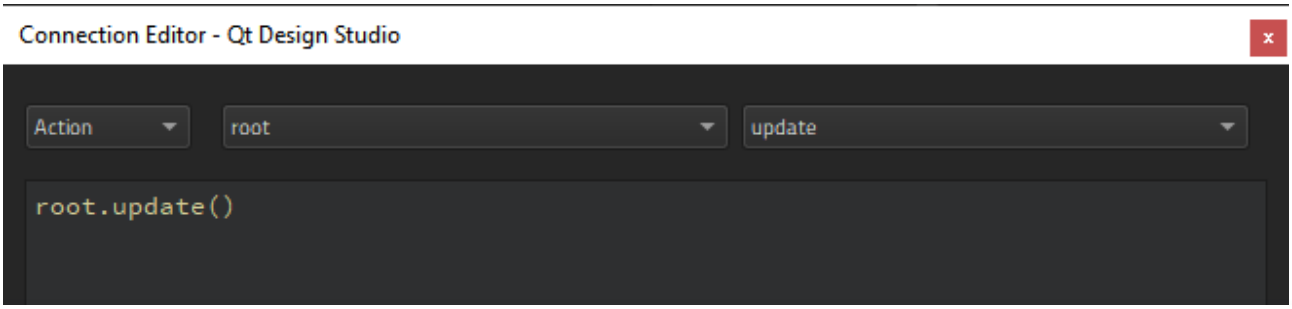
Adding Actions and Assignments

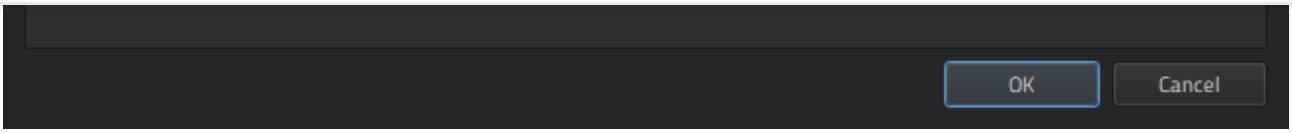
You use the **Connection Editor** to create the JavaScript expressions for *actions* and *assignments*. An *action* connects an component to a signal, whereas an *assignment* fetches property values from another component.

For more information about the logical operators that you can use to construct conditional expressions, see Summary of Logical Operators.

To create JavaScript expressions for actions:

1. Select **Open Connection Editor** in the context menu in **Connections**.

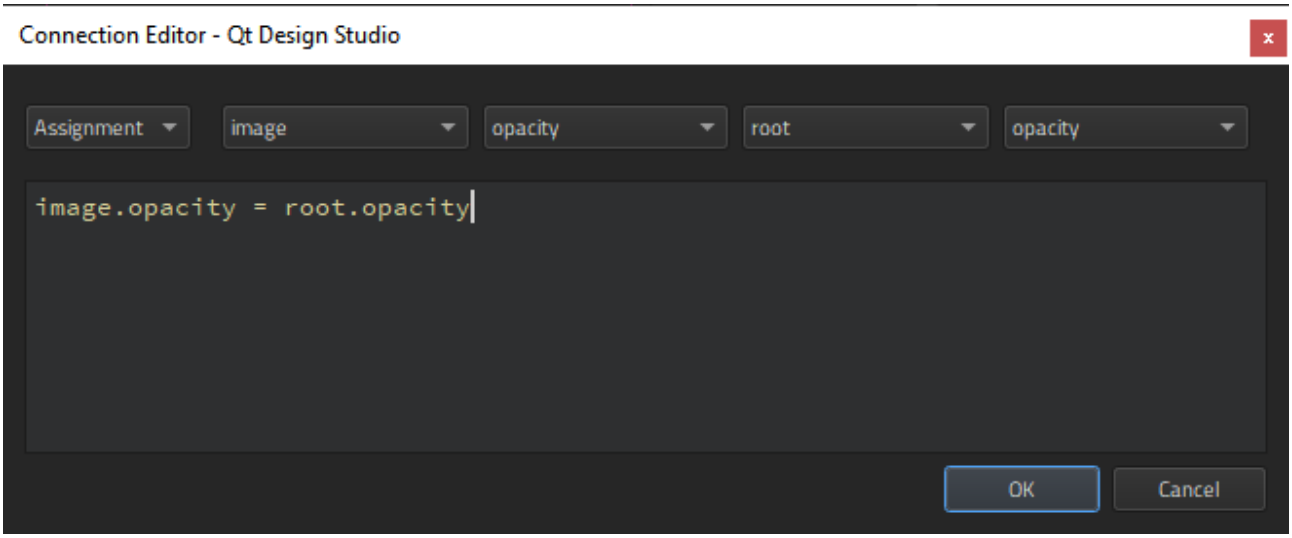




2. Select **Action** as the type of the connections component.
3. Select the component to connect to a signal.
4. Select the action to perform when the signal is emitted.

To create JavaScript expressions for assignments:

1. Select **Open Connection Editor** in the context menu in **Connections**.



2. Select **Assignment** as the type of the connections component.
3. Select the target component for the property assignment.
4. Select the property of the target component to assign a value to.
5. Select the source component for the property assignment.
6. Select the property of the source component to fetch the value from.

‹ Working with Connections

Adding Bindings Between Properties ›

The Qt Company

Contact Us

Qt DOCUMENTATION

Investors
Newsroom
Careers
Office Locations

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

© 2022 The Qt Company

Feedback Sign In