

# UI Files

You can use Qt Design Studio wizards to create UI files that have the filename extension `.ui.qml`. The UI files can be edited in the **2D** view. If you use the **Code** view to add code that is not supported by the **2D** view, Qt Design Studio displays error messages.

The following features are not supported in `.ui.qml` files:

- › JavaScript blocks
- › Other bindings than pure expressions
- › Signal handlers
- › States in other components than the root component
- › Root components that are not derived from `QQuickItem` or `Item`
- › Referencing the parent of the root component

The following components are not supported:

- › Behavior
- › Binding
- › Canvas
- › Shader Effect
- › Timer
- › Transform

## Supported Methods

Qt Design Studio supports most JavaScript functions that are supported by the QML engine, as well as a subset of Qt QML methods.

This section lists the functions that you can use in `.ui.qml` files.

### JavaScript Functions

As a rule of thumb, *pure functions* are supported. They only depend on and modify states of parameters that are within their scope, and therefore always return the same results when given the same parameters. This makes it possible to convert and reformat property bindings without breaking the `.ui.qml` files.

The following JavaScript functions are supported:

- › `concat()`
- › `endsWith()`
- › `includes()`
- › `indexOf()`
- › `isFinite()`
- › `isNaN()`
- › `lastIndexOf()`
- › `substring()`
- › `toExponential()`
- › `toFixed()`
- › `toLocaleLowerCase()`
- › `toLocaleString`
- › `toLocaleUpperCase()`
- › `toLowerCase()`
- › `toPrecision()`
- › `toString()`
- › `toUpperCase()`
- › `valueOf()`

In addition, all functions of the `Math` and `Date` objects are supported.

For more information, see [List of JavaScript Objects and Functions](#).

## Qt QML Methods

Qt Design Studio supports color methods, helper methods for creating objects of specific data types, and translation methods.

The following color methods are supported:

- › `Qt.darker()`
- › `Qt.hsla()`
- › `Qt.hsva()`
- › `Qt.lighter()`
- › `Qt.rgba()`
- › `Qt.tint()`

The following helper methods are supported:

- › `Qt.formatDate()`
- › `Qt.formatDateTime()`
- › `Qt.formatTime()`
- › `Qt.matrix4x4()`

- › `Qt.rect()`
- › `Qt.size()`
- › `Qt.vector2d()`
- › `Qt.vector3d()`
- › `Qt.vector4d()`

The following translation methods are supported:

- › `qsTr()`
- › `qsTranslate()`
- › `qsTranslateNoOp()`
- › `qsTrId()`
- › `qsTrIdNoOp()`
- › `qsTrNoOp()`

**Note:** Do not mix translation methods in a UI file.

For more information about using the methods, see [Qt QML Methods](#).

## Using UI Files

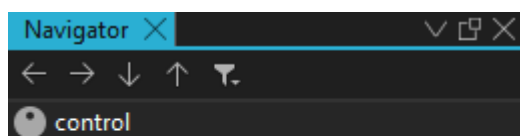
You can edit the UI files in the **2D** and **Text Editor** views. Components that are supposed to be used in code have to be exported as properties:

```
Item {
    width: 640
    height: 480

    property alias button: button

    Button {
        anchors.centerIn: parent
        id: button
        text: qsTr("Press Me")
    }
}
```

The property alias exports the button to the code that uses the form. You can use the **@ (Export)** button in **Navigator** to export a component as a property:



In the UI file where the component is used, you can use the `button` property alias to implement signal handlers, for example. In the following code snippet, the UI file is called *MainForm.ui.qml*:

```
MainForm {  
    anchors.fill: parent  
    button.clicked: messageDialog.show(qsTr("Button pressed"))  
}
```

You can also assign properties or define behavior or transitions.

To move from the **2D** or **Navigator** view directly to the implementation of a component in the .qml file, right-click the component and select **Go to Implementation** in the context menu.

[< Advanced Designer Topics](#)

[Managing Data Collection >](#)



Contact Us

## Company

About Us  
Investors  
Newsroom  
Careers  
Office Locations

## Support

Support Services  
Professional Services  
Partners  
Training

## Licensing

Terms & Conditions  
Open Source  
FAQ

## For Customers

Support Center  
Downloads  
Qt Login  
Contact Us  
Customer Success



- Contribute to Qt
- Forum
- Wiki
- Downloads
- Marketplace

© 2022 The Qt Company

[Feedback](#) [Sign In](#)