

# 用户界面文件

您可以使用 Qt 设计工作室向导创建文件扩展名为 `.ui.qml` 的 UI 文件。可以在 2D 视图中编辑 UI 文件。如果使用“代码”视图添加 2D 视图不支持的代码，Qt Design Studio 将显示错误消息。

`.ui.qml` 文件不支持以下功能：

- 脚本块
- 纯表达式以外的其他绑定
- 信号处理程序
- 根组件以外的其他组件中的状态
- 不是从 `QQuick 项` 或 `项` 派生的根组件
- 引用根组件的父级

不支持以下组件：

- 行为
- 捆绑
- 帆布
- 着色器效果
- 定时器
- 变换

## 支持的方法

Qt 设计工作室支持 QML 引擎支持的大多数 JavaScript 函数，以及 Qt QML 方法的子集。

本节列出了可以在 `.ui.qml` 文件中使用的函数。

### 脚本函数

根据经验，支持纯函数。它们仅依赖于和修改其范围内的参数的状态，因此在给定相同参数时始终返回相同的结果。这样就可以在不破坏 `.ui.qml` 文件的情况下转换和重新设置属性绑定的格式。

支持以下脚本函数：

- `charAt()`
- `charCodeAt()`

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

此外，还支持 和 对象的所有功能。MathDate

有关更多信息，请参见 [JavaScript 对象和函数列表](#)。

## Qt QML 方法

Qt设计工作室支持颜色方法，用于创建特定数据类型对象的帮助器方法和翻译方法。

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.point\(\)](#)
- › [Qt.quaternion\(\)](#)
- › [Qt.rect\(\)](#)

- 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

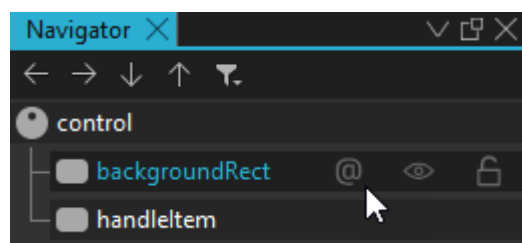
You can edit the UI files in the **2D** and **Text Editor** views. Components that are supposed to be used in code have to be exported as properties:

```
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:



```
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

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



Downloads

Marketplace

© 2022 The Qt Company

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