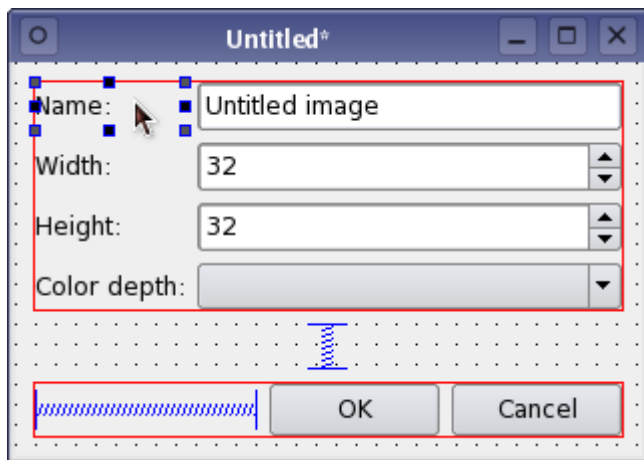


Qt Designer's Widget Editing Mode



In the Widget Editing Mode, objects can be dragged from the main window's widget box to a form, edited, resized, dragged around on the form, and even dragged between forms. Object properties can be modified interactively, so that changes can be seen immediately. The editing interface is intuitive for simple operations, yet it still supports Qt's powerful layout facilities.

To create and edit new forms, open the **File** menu and select **New Form...** or press **Ctrl+N**. Existing forms can also be edited by selecting **Open Form...** from the **File** menu or pressing **Ctrl+O**.

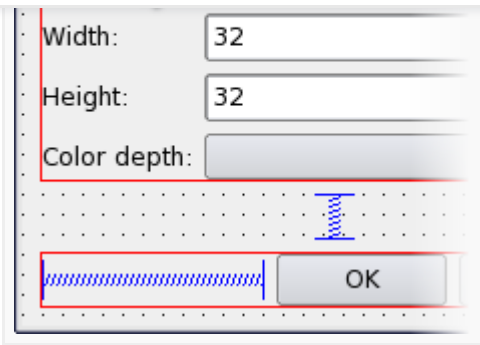
At any point, you can save your form by selecting the **Save Form As...** option from the **File** menu. The UI files saved by *Qt Designer* contain information about the objects used, and any details of signal and slot connections between them.

Editing A Form

By default, new forms are opened in widget editing mode. To switch to Edit mode from another mode, select **Edit Widgets** from the **Edit** menu or press the **F3** key.

Objects are added to the form by dragging them from the main widget box and dropping them in the desired location on the form. Once there, they can be moved around simply by dragging them, or using the cursor keys. Pressing the **Ctrl** key at the same time moves the selected widget pixel by pixel, while using the cursor keys alone make the selected widget snap to the grid when it is moved. Objects can be selected by clicking on them with the left mouse button. You can also use the **Tab** key to change the selection.

The widget box contains objects in a number of different categories, all of which can be placed on the form as required. The only objects that require a little more preparation are the **Container** widgets. These are described in further detail in the [Using Containers in Qt Designer](#) chapter.



corner and the midpoint of each side, indicating that it can be resized.

To select additional objects, hold down the **Control** key and click on them. If more than one object is selected, the current object will be displayed with resize handles of a different color.

To move a widget within a layout, hold down **Shift** and **Control** while dragging the widget. This extends the selection to the widget's parent layout.

Alternatively, objects can be selected in the [Object Inspector](#).

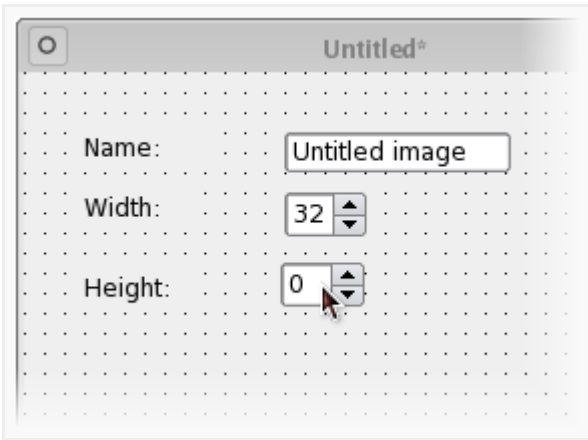
When a widget is selected, normal clipboard operations such as cut, copy, and paste can be performed on it. All of these operations can be done and undone, as necessary.

The following shortcuts can be used:

Action	Shortcut	Description
Cut	Ctrl+X	Cuts the selected objects to the clipboard.
Copy	Ctrl+C	Copies the selected objects to the clipboard.
Paste	Ctrl+V	Pastes the objects in the clipboard onto the form.
Delete	Delete	Deletes the selected objects.
Clone object	Ctrl+drag (leftmouse button)	Makes a copy of the selected object or group of objects.
Preview	Ctrl+R	Shows a preview of the form.

All of the above actions (apart from cloning) can be accessed via both the **Edit** menu and the form's context menu. These menus also provide functions for laying out objects as well as a **Select All** function to select all the objects on the form.

Widgets are not unique objects; you can make as many copies of them as you need. To quickly duplicate a widget, you can clone it by holding down the **Ctrl** key and dragging it. This allows widgets to be copied and placed on the form more quickly than with clipboard operations.



Drag and Drop

Qt Designer makes extensive use of the drag and drop facilities provided by Qt. Widgets can be dragged from the widget box and dropped onto the form.

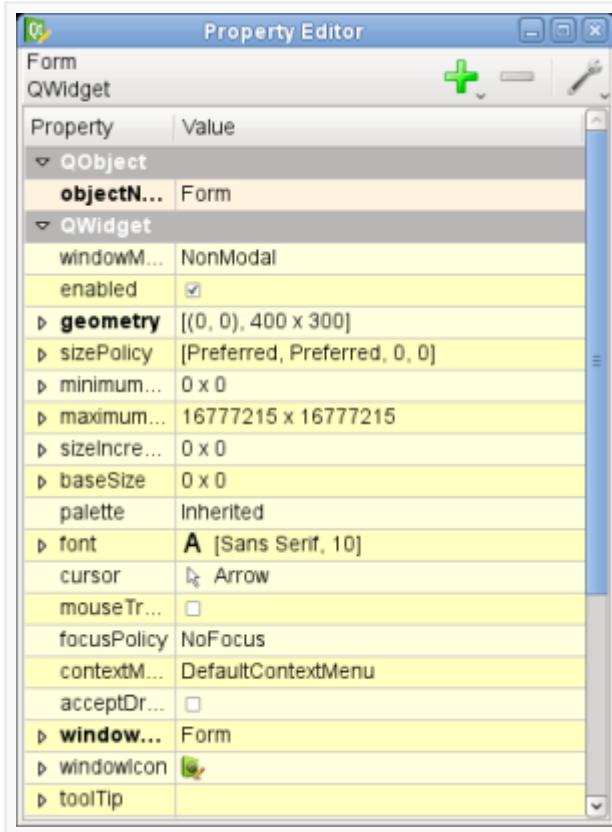
Widgets can also be "cloned" on the form: Holding down **Ctrl** and dragging the widget creates a copy of the widget that can be dragged to a new position.

It is also possible to drop Widgets onto the [Object Inspector](#) to handle nested layouts easily.

Qt Designer allows selections of objects to be copied, pasted, and dragged between forms. You can use this feature to create more than one copy of the same form, and experiment with different layouts in each of them.

The Property Editor

The Property Editor always displays properties of the currently selected object on the form. The available properties



Editing Properties

The property editor uses standard Qt input widgets to manage the properties of objects on the form. Textual properties are shown in line edits, integer properties are displayed in spinboxes, boolean properties are displayed in check boxes, and compound properties such as colors and sizes are presented in drop-down lists of input widgets. Modified properties are indicated with bold labels. To reset them, click the arrow button on the right.

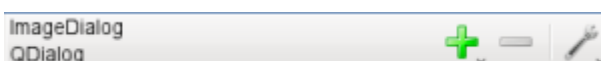
Changes in properties are applied to all selected objects that have the same property.



Certain properties are treated specially by the property editor:

- Compound properties -- properties that are made up of more than one value -- are represented as nodes that can be expanded, allowing their values to be edited.
- Properties that contain a choice or selection of flags are edited via combo boxes with checkable items.
- Properties that allow access to rich data types, such as **QPalette**, are modified using dialogs that open when the properties are edited. **QLabel** and the widgets in the **Buttons** section of the widget box have a **text** property that can also be edited by double-clicking on the widget or by pressing **F2**. *Qt Designer* interprets the backslash (\) character specially, enabling newline (\n) characters to be inserted into the text; the \\ character sequence is used to insert a single backslash into the text. A context menu can also be opened while editing, providing another way to insert special characters and newlines into the text.

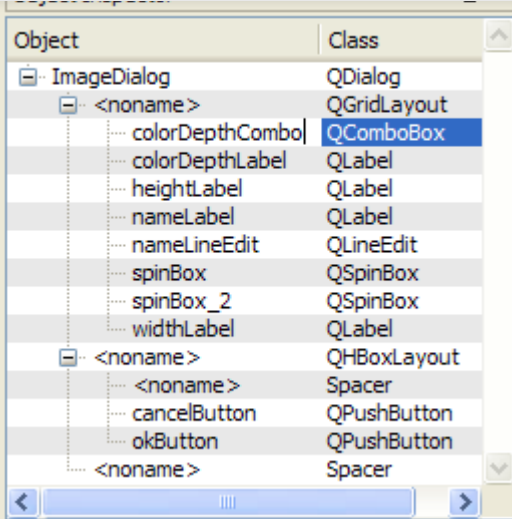
Dynamic Properties

The property editor can also be used to add new **dynamic properties** to both standard Qt widgets and to forms themselves. Since Qt 4.4, dynamic properties are added and removed via the property editor's toolbar, shown below.



To add a dynamic property, click on the **Add** button . To remove it, click on the **Remove** button  instead. You can also sort the properties alphabetically and change the color groups by clicking on the **Configure** button





The **Object Inspector** displays a hierarchical list of all the objects on the form that is currently being edited. To show the child objects of a container widget or a layout, click the handle next to the object label.

Each object on a form can be selected by clicking on the corresponding item in the **Object Inspector**. Right-clicking opens the form's context menu. These features can be useful if you have many overlapping objects. To locate an object in the **Object Inspector**, use **Ctrl+F**.

Since Qt 4.4, double-clicking on the object's name allows you to change the object's name with the in-place editor.

Since Qt 4.5, the **Object Inspector** displays the layout state of the containers. The broken layout icon ###ICON is displayed if there is something wrong with the layouts.

[< Qt Designer's Editing Modes](#)
[Qt Designer's Signals and Slots Editing Mode >](#)

© 2022 The Qt Company Ltd. Documentation contributions included herein are the copyrights of their respective owners. The documentation provided herein is licensed under the terms of the [GNU Free Documentation License version 1.3](#) as published by the Free Software Foundation. Qt and respective logos are [trademarks](#) of The Qt Company Ltd. in Finland and/or other countries worldwide. All other trademarks are property of their respective owners.


[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](#)



Community

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

© 2022 The Qt Company

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