在 Qt 设计器中创建主窗口

*Qt Designer*可用于创建用于不同目的的用户界面，并且它为每个用户界面提供不同类型的表单模板。主窗口模板用于创建具有菜单栏、工具栏和 Dock 小部件的应用程序窗口。

通过打开“**文件**”菜单并选择“**新建窗体...**”选项或按 Ctrl+N 来创建新的主窗口。然后，选择“**主窗口**”模板。默认情况下，此模板提供一个包含菜单栏和工具栏的主应用程序窗口 - 如果不需要，可以删除它们。

如果删除菜单栏，则可以通过从上下文菜单中选择“**创建菜单栏**”选项来创建新的菜单栏，该选项通过在主窗口窗体中单击鼠标右键获得。

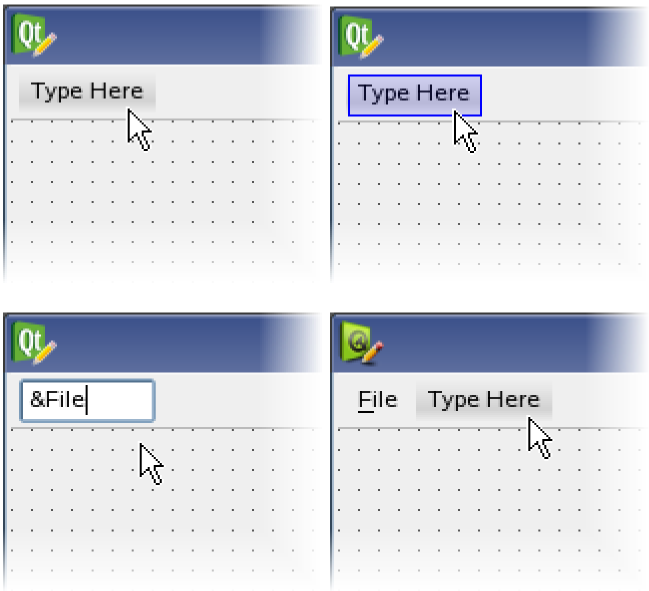
一个应用程序只能有**一个**菜单栏，但**可以有多个**工具栏。

菜单

通过修改“**在此处键入**”占位符将菜单添加到菜单栏中。其中一个始终存在用于编辑目的，并且不会显示在预览或完成的窗口中。

创建后，可以使用属性编辑器访问菜单的属性，并且可以为此目的通过对象检查器访问每个菜单。

可以通过在菜单栏中的标签上打开上下文菜单，然后选择删除菜单“menu_name”**来删除现有菜单**。

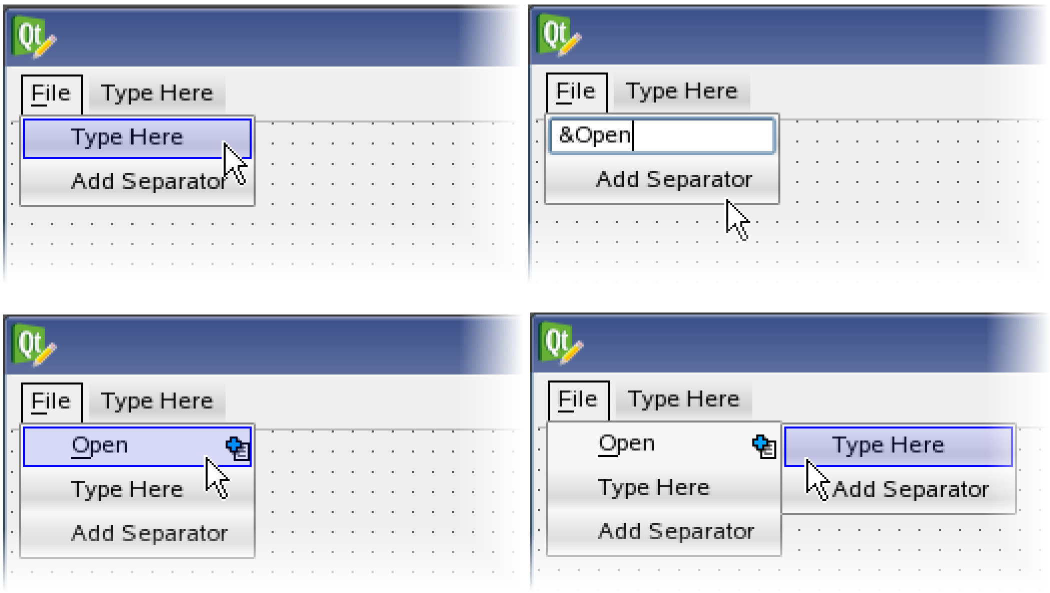


创建菜单

双击占位符项以开始编辑。可以使用行编辑显示的菜单文本可以修改。

菜单也可以在菜单栏中重新排列，只需将它们拖放到首选位置即可。垂直红线表示菜单的插入位置。

菜单可以包含任意数量的条目和分隔符，并且可以嵌套到所需的深度。通过以常规方式导航菜单结构，可以向菜单添加新条目。



创建菜单项

双击"**在此处键入**"占位符开始编辑，或双击"**添加分隔符**"在菜单中最后一个条目后插入新的分隔线。

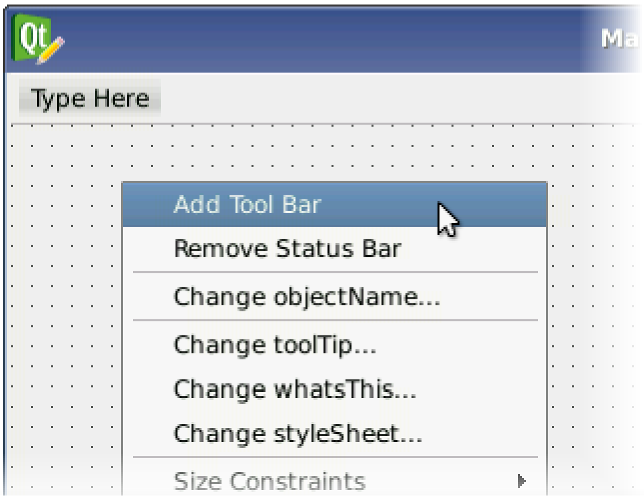
菜单项的文本使用行编辑显示，并且可以修改。

插入新条目所需的文本，可以选择使用 & 符号字符 （&） 标记要用作条目助记符的字母。

按"**返回**"或"**回车键**"接受新文本，或按 **Esc 键**拒绝新文本。为此菜单项创建的操作将通过操作编辑器进行访问，并且可以在其中设置任何关联的键盘快捷键。

就像菜单一样，只需将条目拖放到首选位置即可移动条目。当一个条目被拖到一个关闭的菜单上时，菜单将打开，允许它插入到那里。由于菜单项基于操作，因此也可以将它们放到工具栏上，在那里它们将显示为工具栏按钮。

工具栏



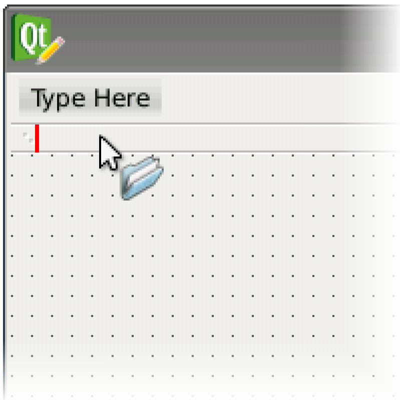
Creating and Removing a Toolbar

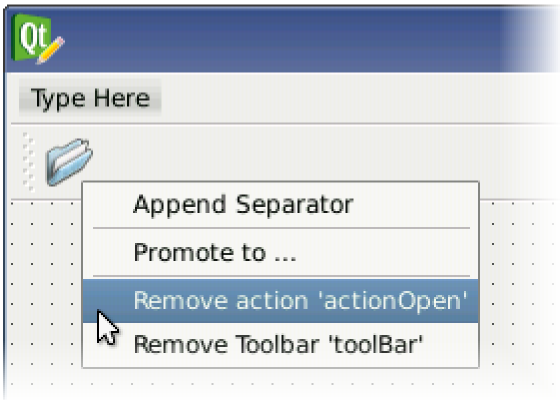
Toolbars are added to a main window in a similar way to the menu bar: Select the **Add Tool Bar** option from the form's context menu. Alternatively, if there is an existing toolbar in the main window, you can click the arrow on its right end to create a new toolbar.

Toolbars are removed from the form via an entry in the toolbar's context menu.

Adding and Removing Toolbar Buttons

Toolbar buttons are created as actions in the Action Editor and dragged onto the toolbar. Since actions can be represented by menu entries and toolbar buttons, they can be moved between menus and toolbars.



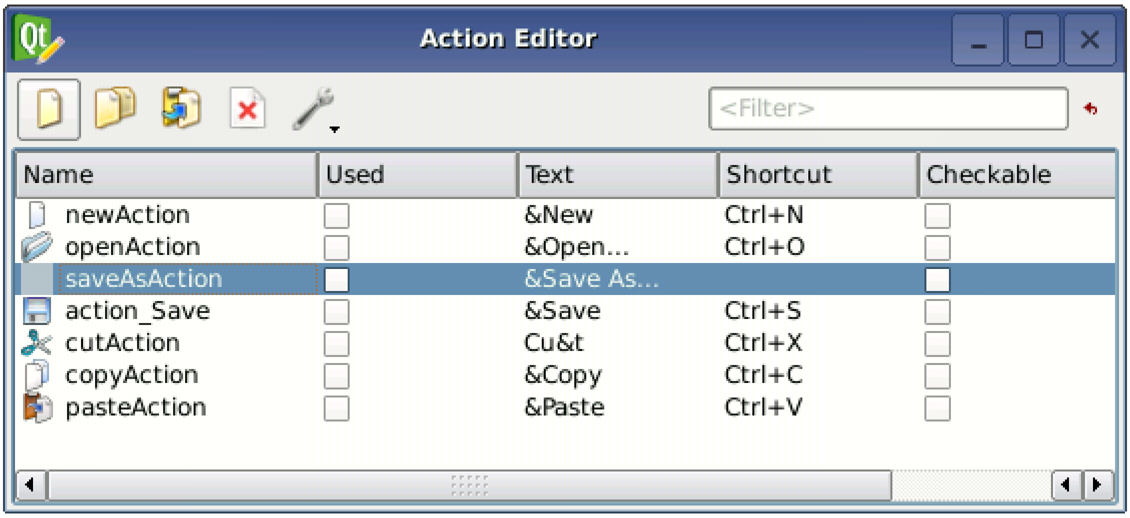


To share an action between a menu and a toolbar, drag its icon from the action editor to the toolbar rather than from the menu where its entry is located. See Adding an Action for more information about this process.

Toolbar buttons are removed via the toolbar's context menu.

Actions

With the menu bar and the toolbars in place, it's time to populate them with actions. New actions for both menus and toolbars are created in the action editor window, simplifying the creation and management of actions.



Name	Used	Text	Shortcut	Checkable
newAction	□	&New	Ctrl+N	□
openAction	□	&Open...	Ctrl+O	□
saveAsAction	□	&Save As...		□
action_Save	□	&Save	Ctrl+S	□
cutAction	□	Cu&t	Ctrl+X	□
copyAction	□	&Copy	Ctrl+C	□
pasteAction	□	&Paste	Ctrl+V	□

The Action Editor

Enable the action editor by opening the **Tools** menu, and switching on the **Action Editor** option.

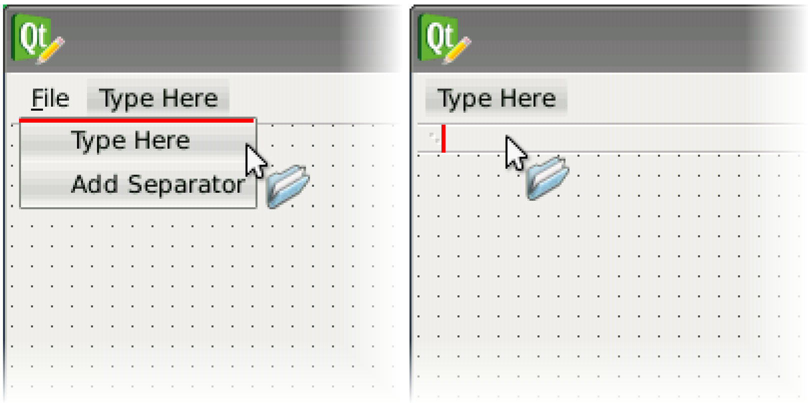
Qt Designer's action editor can be viewed in the classic **Icon View** and **Detailed View**. The screenshot below shows the action editor in **Detailed View**. You can also copy and paste actions between menus, toolbars and forms.

Creating an Action

To create an action, use the action editor's **New** button, which will then pop up an input dialog. Provide the new action with a **Text** -- this is the text that will appear in a menu entry and as the action's tooltip. The text is also automatically added to an "action" prefix, creating the action's **Object Name**.

In addition, the dialog provides the option of selecting an **Icon** for the action, as well as removing the current icon.

Once the action is created, it can be used wherever actions are applicable.



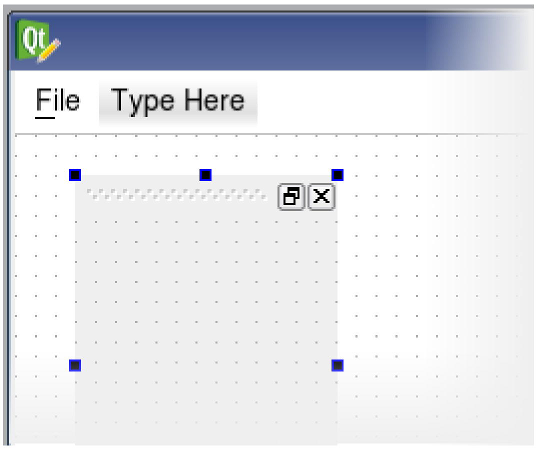
Adding an Action

To add an action to a menu or a toolbar, simply press the left mouse button over the action in the action editor, and drag it to the preferred location.

Qt Designer provides highlighted guide lines that tell you where the action will be added. Release the mouse button to add the action when you have found the right spot.

Dock Widgets

Dock widgets are container widgets as well. They can be added to a form by dropping them onto the desired dock area.



Adding a Dock Widget

docked property to place it in a dock area.

Note that it is sometimes easier to configure a dock widget if it is added to a form before a layout is applied to the central widget. For example, it is possible to undock it and resize it, making it more convenient to add child widgets.

Dock widgets can be optionally floated as independent tool windows. Hence, it is useful to give them window titles by setting their windowTitle property. This also helps to identify them on the form.

< Using Containers in Qt Designer

Editing Resources with Qt Designer >

Qt The Qt Company

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

Feedback Sign In