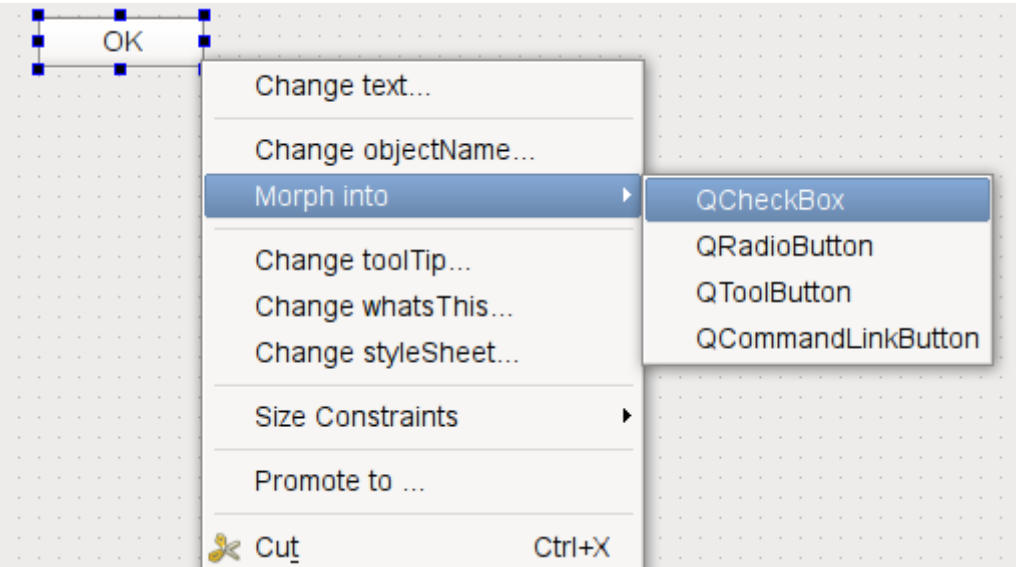


# Using Containers in Qt Designer

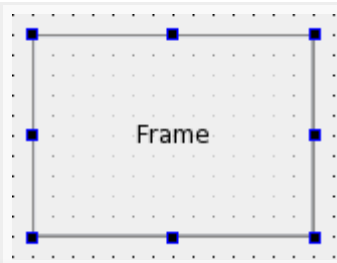
Container widgets provide high level control over groups of objects on a form. They can be used to perform a variety of functions, such as managing input widgets, providing paged and tabbed layouts, or just acting as decorative containers for other objects.

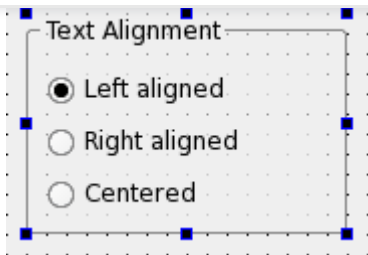


*Qt Designer* provides visual feedback to help you place objects inside your containers. When you drag an object from the widget box (or elsewhere) on the form, each container will be highlighted when the cursor is positioned over it. This indicates that you can drop the object inside, making it a child object of the container. This feedback is important because it is easy to place objects close to containers without actually placing them inside. Both widgets and spacers can be used inside containers.

Stacked widgets, tab widgets, and toolboxes are handled specially in *Qt Designer*. Normally, when adding pages (tabs, pages, compartments) to these containers in your own code, you need to supply existing widgets, either as placeholders or containing child widgets. In *Qt Designer*, these are automatically created for you, so you can add child objects to each page straight away.

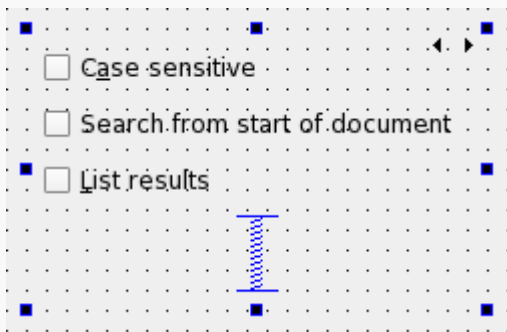
Each container typically allows its child objects to be arranged in one or more layouts. The type of layout management provided depends on each container, although setting the layout is usually just a matter of selecting the container by clicking it, and applying a layout. The table below shows a list of available containers.

	<p><b>Frames</b></p> <p>Frames are used to enclose and group widgets, as well as to provide decoration. They are used as the foundation for more complex containers, but they can also be used as placeholders in forms.</p> <p>The most important properties of frames are <code>frameShape</code>, <code>frameShadow</code>, <code>lineWidth</code>, and</p>
---	--



### Group Boxes

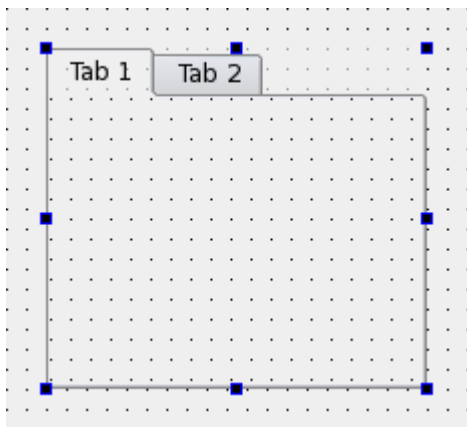
Group boxes are usually used to group together collections of checkboxes and radio buttons with similar purposes. Among the significant properties of group boxes are **title**, **flat**, **checkable**, and **checked**. These are demonstrated in the [Group Box](#) example, and described in the [QGroupBox](#) class documentation. Each group box can contain its own layout, and this is necessary if it contains other widgets. To add a layout to the group box, click inside it and apply the layout as usual.



### Stacked Widgets

Stacked widgets are collections of widgets in which only the topmost layer is visible. Control over the visible layer is usually managed by another widget, such as combobox, using signals and slots.

*Qt Designer* shows arrows in the top-right corner of the stack to allow you to see all the widgets in the stack when designing it. These arrows do not appear in the preview or in the final component. To navigate between pages in the stack, select the stacked widget and use the **Next Page** and **Previous Page** entries from the context menu. The **Insert Page** and **Delete Page** context menu options allow you to add and remove pages.



### Tab Widgets

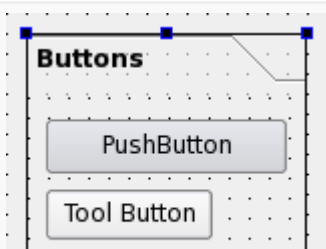
Tab widgets allow the developer to split up the contents of a widget into different labelled sections, only one of which is displayed at any given time. By default, the tab widget contains two tabs, and these can be deleted or renamed as required. You can also add additional tabs.

To delete a tab:

- › Click on its label to make it the current tab.
- › Select the tab widget and open its context menu.
- › Select **Delete Page**.

To add a new tab:

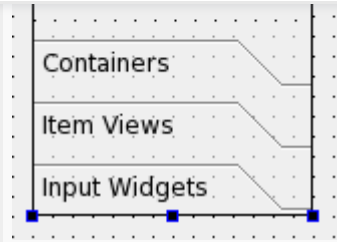
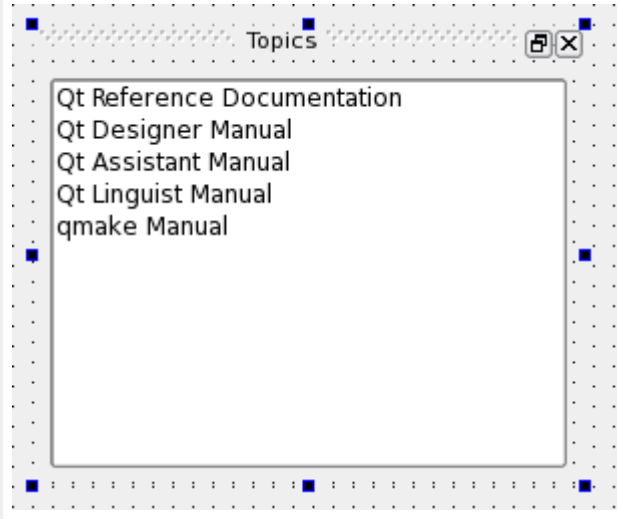
- › Select the tab widget and open its context menu.
- › Select **Insert Page**.
- › You can add a page before or after the *current* page. *Qt Designer* will create a new widget for that particular tab and insert it into the tab widget.
- › You can set the title of the current tab by changing the `currentTabText` property in the **Property Editor**.



### ToolBox Widgets

Toolbox widgets provide a series of pages or compartments in a toolbox. They are handled in a way similar to stacked widgets.

To rename a page in a toolbox, make the toolbox your current pange and change its `currentItemText` property from the **Property Editor**.

	To delete a page, select <b>Delete Page</b> from the toolbox widget's context menu.
	<b>Dock Widgets</b> Dock widgets are floating panels, often containing input widgets and more complex controls, that are either attached to the edges of the main window in "dock areas", or floated as independent tool windows. Although dock widgets can be added to any type of form, they are typically used with forms created from the <a href="#">main window template</a> .

< Qt Designer's Tab Order Editing Mode

Creating Main Windows in Qt Designer >

© 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

Licensing

- Terms & Conditions
- Open Source
- FAQ

For Customers



Partners

Training

Qt Login

Contact Us

Customer Success

Community

Contribute to Qt

Forum

Wiki

Downloads

Marketplace

© 2022 The Qt Company

Feedback Sign In