

选项卡式小部件QTabWidget

[TOC]

QTabWidget提供一堆选项卡式小部件。

详细介绍



选项卡式部件提供一个选项卡栏和一个用于显示与每个选项卡相关的页面的页面区域。

默认情况下，选项卡栏显示在页面区域上方，可以使用不同的配置，设置TabPosition属性。

每个选项卡都与一个不同的小部件相关联，这个小部件成为页面。

页面区域中只显示当前页面，所有其他页面都被隐藏。可以通过单击选项卡或按下Alt+字母快捷键来显示不同的页面。

一般使用QTabWidget的方法如下操作：

1. 创建一个QTabWidget对象；
2. 为选项卡中的每个页面创建一个QWidget对象，该对象不要指定父窗口；
3. 将子窗口小部件插入页面窗口小部件，使用布局将其正常定位；
4. 调用addTab()或insertTab()函数将页面小部件放入选项卡小部件中，为每个选项卡页面提供一个合适的标签和一个可选的键盘快捷键。

示例如下：

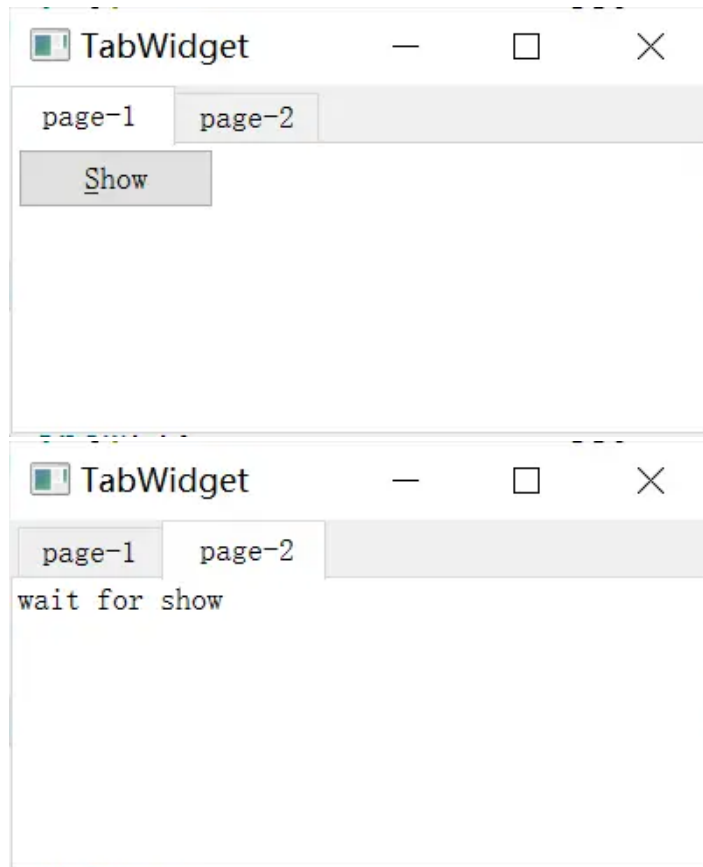
```
1 // 1. 创建一个QTabWidget对象
2 QTabWidget *tabWidget = new QTabWidget(this);
3 // 2. 为选项卡中的页面创建QWidget对象，创建2个选项卡
4 QWidget *page1 = new QWidget;
5 QWidget *page2 = new QWidget;
6 //3. 将子窗口小部件插入到页面中
7 // page1页面插入一个按钮，page2页面插入一个标签
8 QPushButton *btn = new QPushButton(tr("&Show"), page1);
```

```

9     QLabel *label = new QLabel(tr("wait for show"), page2);
10    // 4. 将页面添加到QTabWidget对象中，并为每个选项卡提供一个合适的标签和可选的键盘
11    tabWidget->addTab(page1, tr("page-1"));
12    tabWidget->addTab(page2, tr("page-2"));

```

显示如下：



选项卡的位置由tabPosition属性定义，默认在North；形状由tabShape定义，默认是Rounded。

当选择一个页面时会发送currentChanged()信号。

当前页面的索引调用currentIndex()获取，当前页面的部件调用currentWidget()函数获取。widget()函数获取给定索引的页面部件的指针，indexOf()函数查找部件的索引位置。setCurrentWidget()或setCurrentIndex()显示特定页面。

可以调用setTabText()或setTabIcon()函数更改选项卡的文本和图标；可以使用removeTab()函数删除选项卡及其关联页面。

调用setTabEnabled()可以启用或禁用选项卡页面。

- 如果启用某个选项卡页面，会正常绘制该选项卡，并且可以选择该选项卡；
- 如果禁用某个选项卡页面，以不同方式绘制该选项卡，且用户无法选择此选项卡。

需要注意的是，即使禁用了某个页签，页面仍然是可见的。

选项卡是拆分复杂对话框的一种很好的方法。另一种选择是使用QStackedWidget，为其提供一些在页面之间导航的方法，比如QToolBar或QListWidget。

QTabWidget中的大部分功能由QToolBar(位于顶部，提供选项卡)和QStackedWidget(大部分区域，组织各个页面)提供。

常用属性

除了继承自QWidget的属性外，QTabWidget类提供了一些常用的属性，如下：

属性	类型及说明
count	int，选项卡栏中的选项卡个数
currentIndex	int，当前页面的索引，-1表示没有当前部件
documentModel	bool，标志选项卡部件是否以适合文档页面的模式呈现。 true不会呈现选项卡部件框架，此时适用于显示文档类型
elideMode	Qt::TextElideMode，如何消隐选项卡栏中的文本。 控制当没有足够的空间显示给定选项卡栏大小的项目时，如何对其进行消隐。
iconSize	QSize，选项卡栏中图标的大小。
movable	bool，用户是否可以在选项卡区域内移动选项卡。
tabBarAutoHide	bool，如果为true，选项卡栏包含的选项卡少于2个时，该选项卡栏将自动隐藏。默认情况下，该属性为false。
tabPosition	TabPosition，选项卡栏的位置
tabShape	TabShape，选项卡小部件中选项卡的形状。
tabsClosable	bool，是否将关闭按钮自动添加到每个选项卡。
usesScrollButtons	bool，当选项卡栏有多个按钮时，此属性决定是否应使用按钮滚动选项卡。 当选项卡栏中的选项卡太多而无法容纳其大小时，选项卡栏可以选择扩展其大小，也可以选择添加允许滚动选项卡的按钮。

信号

QTabWidget提供的信号有：

信号	说明
`currentChanged()`	页面索引发生变化时发送该信号
`tabBarClicked()`	用户单击索引处的选项卡时，发出该信号。
`tabBarDoubledClicked()`	用户双击索引处的选项卡时发出该信号。
`tabCloseRequested()`	单击选项卡上的关闭按钮时发出此信号。

常用函数

除了与获取和设置属性相关的函数外，QTabWidget还提供了一些有用的函数：

- 插入页面函数

`addTab()`，如果在`show()`之后调用该函数，布局系统将尝试根据其小部件层次结构的变化进行调整，可能导致闪烁。为了防止这种情况发生，可以在更改之前将`updateEnabled`属性设置为`false`，在完成更改之后将属性设置为`true`，使小部件再次接收绘制事件。

`insertTab()`

- 删除或清空

`clear()`，移除所有的页面，但是不删除它们。

`removeTab()`，移除指定索引处的页面，部件本身不会被删除。

- 选项卡栏相关

`tabBar()`，获取QTabBar类对象。

官方示例Tab Dialog

官方示例Tab Dialog显示了如何使用QTabWidget类构建选项卡对话框。

选项卡对话框示例由一个TabDialog类组成，该类提供三个选项卡，每个选项卡都包含有关特定文件的信息，以及两个用于接受或拒绝对话框内容的标准按钮。