
pyqtribbon

Release 0.1.5

WANG Hailin

Jul 23, 2022

CONTENTS:

1	Getting Started	3
1.1	Installation	3
2	The Ribbon Bar	5
2.1	Introduction	5
2.2	Definitions of Ribbon Elements	6
2.3	Ribbon Elements in PyQtRibbon	6
3	Tutorials	7
3.1	Using the RibbonBar	7
3.2	Using the RibbonMainWindow	8
4	API References	11
4.1	Main Window	11
4.2	Ribbon Bar	11
4.3	Ribbon Title	16
4.4	Ribbon Category	20
4.5	Ribbon Panel	24
4.6	Ribbon Gallery	36
4.7	Ribbon Tool Button	38
4.8	Ribbon Separator	39
5	Indices and tables	41
	Index	43

PyQtRibbon is a Qt-based application framework for building user interfaces.

- GitHub Repository: github.com/haibiliin/pyqtribbon.
- PyPI: pypi.org/project/pyqtribbon.
- Documentation: pyqtribbon.haibiliin.com.
- Read the Docs: readthedocs.org/projects/pyqtribbon.

GETTING STARTED

1.1 Installation

pyqtribbon is distributed to [PyPI](#), you can use pip to install it:

```
pip install pyqtribbon
```

You can also install the package from source:

```
pip install git+https://github.com/haiiliin/pyqtribbon.git@main
```


THE RIBBON BAR

2.1 Introduction

The ribbon is first introduced by Microsoft in the 2000's. It is a toolbar with a tabbed interface. According to [Microsoft](#):

Note: A ribbon is a user interface (UI) element that organizes commands into logical groups. These groups appear on separate tabs in a strip across the top of the window. The ribbon replaces the menu bar and toolbars. A ribbon can significantly improve application usability. For more information, see [Ribbons](#). The following illustration shows a ribbon. A ribbon can significantly improve application usability. For more information, see [Ribbons](#). The following illustration shows a ribbon.

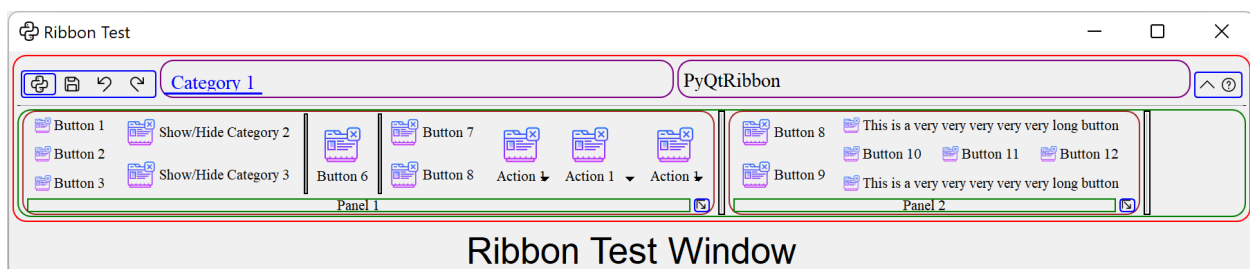


2.2 Definitions of Ribbon Elements



- **Application button**: The button that appears on the upper-left corner of a ribbon. The Application button replaces the File menu and is visible even when the ribbon is minimized. When the button is clicked, a menu that has a list of commands is displayed.
- **Quick Access toolbar**: A small, customizable toolbar that displays frequently used commands.
- **Category**: The logical grouping that represents the contents of a ribbon tab.
- **Category Default button**: The button that appears on the ribbon when the ribbon is minimized. When the button is clicked, the category reappears as a menu.
- **Panel**: An area of the ribbon bar that displays a group of related controls. Every ribbon category contains one or more ribbon panels.
- **Ribbon elements**: Controls in the panels, for example, buttons and combo boxes. To see the various controls that can be hosted on a ribbon, see RibbonGadgets Sample: Ribbon Gadgets Application.

2.3 Ribbon Elements in PyQtRibbon



3.1 Using the RibbonBar

RibbonBar is a class that implements a ribbon bar, you can use it to create a ribbon bar.

```
import sys

from PyQt5 import QtWidgets, QtGui, QtCore

from ribbon import RibbonBar
from ribbon.utils import data_file_path

if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    app.setFont(QtGui.QFont("Times New Roman", 8))

    # Central widget
    window = QtWidgets.QMainWindow()
    window.setWindowIcon(QtGui.QIcon(data_file_path("icons/python.png")))
    centralWidget = QtWidgets.QWidget()
    window.setCentralWidget(centralWidget)
    layout = QtWidgets.QVBoxLayout(centralWidget)

    # Ribbon bar
    ribbonbar = RibbonBar()
    category = ribbonbar.addCategory("Category 1")
    panel = category.addPanel("Panel 1")
    panel.addLargeButton("A Large Button", QtGui.QIcon(data_file_path("icons/python.png")))
    panel.addMediumButton("A Medium Button", QtGui.QIcon(data_file_path("icons/python.png")))
    panel.addMediumButton("A Medium Button", QtGui.QIcon(data_file_path("icons/python.png")))
    panel.addSmallButton("A Medium Button", QtGui.QIcon(data_file_path("icons/python.png")))
    panel.addSmallButton("A Small Button", QtGui.QIcon(data_file_path("icons/python.png")))
    panel.addSmallButton("A Small Button", QtGui.QIcon(data_file_path("icons/python.png")))
```

(continues on next page)

(continued from previous page)

```

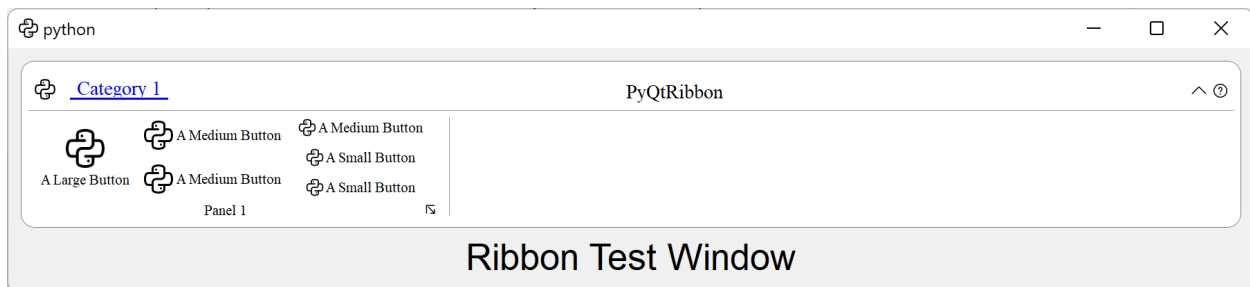
# Display a label in the main window
label = QtWidgets.QLabel("Ribbon Test Window")
label = QtWidgets.QLabel("Ribbon Test Window")
label.setFont(QtGui.QFont("Arial", 20))
label.setAlignment(QtCore.Qt.AlignCenter)

# Add the ribbon bar and label to the layout
layout.addWidget(ribbonbar, 0)
layout.addWidget(label, 1)

# Show the window
window.resize(1800, 350)
window.show()
sys.exit(app.exec_())

```

It would be rendered as follows:



3.2 Using the RibbonMainWindow

Alternatively, you can use the `RibbonMainWindow` which is a `QMainWindow` that has a ribbon bar.

```

import sys

from PyQt5 import QtWidgets, QtGui, QtCore

from ribbon import RibbonMainWindow
from ribbon.utils import data_file_path

if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    app.setFont(QtGui.QFont("Times New Roman", 8))

    # Main window
    window = RibbonMainWindow()
    window.setWindowIcon(QtGui.QIcon(data_file_path("icons/python.png")))

    # Ribbon bar
    ribbonbar = window.ribbonBar()
    category = ribbonbar.addCategory("Category 1")
    panel = category.addPanel("Panel 1")

```

(continues on next page)

(continued from previous page)

```
panel.addLargeButton("A Large Button", QtGui.QIcon(data_file_path("icons/python.png
↪")))
panel.addMediumButton("A Medium Button", QtGui.QIcon(data_file_path("icons/python.png
↪")))
panel.addMediumButton("A Medium Button", QtGui.QIcon(data_file_path("icons/python.png
↪")))
panel.addSmallButton("A Medium Button", QtGui.QIcon(data_file_path("icons/python.png
↪")))
panel.addSmallButton("A Small Button", QtGui.QIcon(data_file_path("icons/python.png
↪")))
panel.addSmallButton("A Small Button", QtGui.QIcon(data_file_path("icons/python.png
↪")))

# Display a label in the main window
label = QtWidgets.QLabel("Ribbon Test Window")
label.setFont(QtGui.QFont("Arial", 20))
label.setAlignment(QtCore.Qt.AlignCenter)

# Add the ribbon bar and label to the layout
window.layout().addWidget(ribbonbar, 0)
window.layout().addWidget(label, 1)

# Show the window
window.resize(1800, 350)
window.show()
sys.exit(app.exec_())
```


API REFERENCES

4.1 Main Window

class ribbon.mainwindow.**RibbonMainWindow**(parent=None)

Main window with ribbon bar.

Methods

<i>layout()</i>	Return the main layout.
<i>ribbonBar()</i>	Return the ribbon bar.

layout() → QVBoxLayout
Return the main layout.

ribbonBar() → *RibbonBar*
Return the ribbon bar.

4.2 Ribbon Bar

class ribbon.ribbonbar.**RibbonBar**(title='PyQtRibbon', parent=None)

The RibbonBar class is the top level widget that contains the ribbon.

Methods

<i>addApplicationOptionAction</i> (action)	Add a display option to the ribbon.
<i>addCategory</i> (title[, style, color])	Add a new category to the ribbon.
<i>addContextCategory</i> (title[, color])	Add a new context category to the ribbon.
<i>addNormalCategory</i> (title)	Add a new category to the ribbon.
<i>addQuickAccessButton</i> (button)	Add a button to the quick access bar.
<i>addRightToolButton</i> (button)	Add a widget to the right button bar.
<i>applicationOptionButton</i> ()	Return the application button.
<i>categories</i> ()	Return a list of categories of the ribbon.
<i>collapseRibbonButton</i> ()	Return the collapse ribbon button.

continues on next page

Table 1 – continued from previous page

<code>hideContextCategory(category)</code>	Hide the given category, if it is not a context category, nothing happens.
<code>hideRibbon()</code>	Hide the ribbon.
<code>minimumSizeHint()</code>	Return the minimum size hint of the widget.
<code>quickAccessToolBar()</code>	Return the quick access toolbar of the ribbon.
<code>removeCategory(category)</code>	Remove a category from the ribbon.
<code>removeCollapseButton()</code>	Remove the min button from the ribbon.
<code>removeHelpButton()</code>	Remove the help button from the ribbon.
<code>ribbonHeight()</code>	Get the total height of the ribbon.
<code>ribbonVisible()</code>	Get the visibility of the ribbon.
<code>rightToolBar()</code>	Return the right toolbar of the ribbon.
<code>setApplicationIcon(icon)</code>	Set the application icon.
<code>setCollapseButtonIcon(icon)</code>	Set the icon of the min button.
<code>setCurrentCategory(category)</code>	Set the current category.
<code>setHelpButtonIcon(icon)</code>	Set the icon of the help button.
<code>setQuickAccessButtonHeight([height])</code>	Set the height of the quick access buttons.
<code>setRibbonHeight(height)</code>	Set the total height of the ribbon.
<code>setRibbonStyle(style)</code>	Set the style of the ribbon.
<code>setRibbonVisible(visible)</code>	Set the visibility of the ribbon.
<code>setRightToolBarHeight([height])</code>	Set the height of the right buttons.
<code>setTabBarHeight([height])</code>	Set the height of the tab bar.
<code>setTitle(title)</code>	Set the title of the ribbon.
<code>showContextCategory(category)</code>	Show the given category, if it is not a context category, nothing happens.
<code>showRibbon()</code>	Show the ribbon.
<code>tabBar()</code>	Return the tab bar of the ribbon.
<code>tabBarHeight()</code>	Get the height of the tab bar.
<code>tabRect(category)</code>	Get the rectangle of the tab of the given category.
<code>title()</code>	Return the title of the ribbon.

helpButtonClicked	
-------------------	--

addApplicationOptionAction(*action: QAction*)

Add a display option to the ribbon.

Parameters

action – The action of the display option.

addCategory(*title: str, style=RibbonCategoryStyle.Normal, color: Optional[QColor] = None*) → Union[*RibbonNormalCategory, RibbonContextCategory*]

Add a new category to the ribbon.

Parameters

- **title** – The title of the category.
- **style** – The button style of the category.
- **color** – The color of the context category, only used if style is Context, if None, the default color will be used.

Returns

The newly created category.

addContextCategory(*title: str, color: Optional[QColor] = None*) → *RibbonContextCategory*

Add a new context category to the ribbon.

Parameters

- **title** – The title of the category.
- **color** – The color of the context category, if None, the default color will be used.

Returns

The newly created category.

addNormalCategory(*title: str*) → *RibbonNormalCategory*

Add a new category to the ribbon.

Parameters

title – The title of the category.

Returns

The newly created category.

addQuickAccessButton(*button: QToolButton*)

Add a button to the quick access bar.

Parameters

button – The button to add.

addRightToolButton(*button: QToolButton*)

Add a widget to the right button bar.

Parameters

button – The button to add.

applicationOptionButton()

Return the application button.

categories() → List[*RibbonCategory*]

Return a list of categories of the ribbon.

Returns

A list of categories of the ribbon.

collapseRibbonButton() → *QToolButton*

Return the collapse ribbon button.

Returns

The collapse ribbon button.

helpButtonClicked

Signal, The help button was clicked.

hideContextCategory(*category: RibbonContextCategory*)

Hide the given category, if it is not a context category, nothing happens.

Parameters

category – The category to hide.

hideRibbon()

Hide the ribbon.

minimumSizeHint() → QSize

Return the minimum size hint of the widget.

Returns

The minimum size hint.

quickAccessToolBar() → QToolBar

Return the quick access toolbar of the ribbon.

Returns

The quick access toolbar of the ribbon.

removeCategory(category: RibbonCategory)

Remove a category from the ribbon.

Parameters

category – The category to remove.

removeCollapseButton()

Remove the min button from the ribbon.

removeHelpButton()

Remove the help button from the ribbon.

ribbonHeight() → int

Get the total height of the ribbon.

Returns

The height of the ribbon.

ribbonVisible() → bool

Get the visibility of the ribbon.

Returns

True if the ribbon is visible, False otherwise.

rightToolBar() → QToolBar

Return the right toolbar of the ribbon.

Returns

The right toolbar of the ribbon.

setApplicationIcon(icon: QIcon)

Set the application icon.

Parameters

icon – The icon to set.

setCollapseButtonIcon(icon: QIcon)

Set the icon of the min button.

Parameters

icon – The icon to set.

setCurrentCategory(category: RibbonCategory)

Set the current category.

Parameters

category – The category to set.

setHelpButtonIcon(*icon: QIcon*)

Set the icon of the help button.

Parameters

icon – The icon to set.

setQuickAccessButtonHeight(*height: int = 40*)

Set the height of the quick access buttons.

Parameters

height – The height to set.

setRibbonHeight(*height: int*)

Set the total height of the ribbon.

Parameters

height – The height to set.

setRibbonStyle(*style: RibbonStyle*)

Set the style of the ribbon.

Parameters

style – The style to set.

setRibbonVisible(*visible: bool*)

Set the visibility of the ribbon.

Parameters

visible – True to show the ribbon, False to hide it.

setRightToolBarHeight(*height: int = 24*)

Set the height of the right buttons.

Parameters

height – The height to set.

setTabBarHeight(*height: int = 50*)

Set the height of the tab bar.

Parameters

height – The height to set.

setTitle(*title: str*)

Set the title of the ribbon.

Parameters

title – The title to set.

showContextCategory(*category: RibbonContextCategory*)

Show the given category, if it is not a context category, nothing happens.

Parameters

category – The category to show.

showRibbon()

Show the ribbon.

tabBar()

Return the tab bar of the ribbon.

Returns

The tab bar of the ribbon.

tabBarHeight() → int

Get the height of the tab bar.

Returns

The height of the tab bar.

tabRect(category: [RibbonCategory](#)) → QRect

Get the rectangle of the tab of the given category.

Parameters

category – The category to get the tab rectangle of.

Returns

The rectangle of the tab.

title()

Return the title of the ribbon.

Returns

The title of the ribbon.

4.3 Ribbon Title

4.3.1 RibbonApplicationButton

class ribbon.titlewidget.**RibbonApplicationButton**

Application button in the ribbon bar.

4.3.2 RibbonTabBar

class ribbon.tabbar.**RibbonTabBar**(parent=None)

The TabBar for the title widget.

Methods

<i>addTab</i> (text[, color])	Add a new tab to the tab bar.
<i>currentTabColor</i> ()	Current tab color
<i>indexOf</i> (tabName)	Return the index of the tab with the given name.
<i>paintEvent</i> (a0)	Paint the tab bar.

addTab(text: str, color: Optional[QColor] = None) → int

Add a new tab to the tab bar.

Parameters

- **text** – The text of the tab.
- **color** – The color of the tab.

Returns

The index of the tab.

currentTabColor() → QColor

Current tab color

Returns

Current tab color

indexOf(tabName: str) → int

Return the index of the tab with the given name.

Parameters

tabName – The name of the tab.

Returns

The index of the tab.

paintEvent(a0: QPaintEvent) → None

Paint the tab bar.

4.3.3 RibbonTitleLabel

class ribbon.titlewidget.RibbonTitleLabel

Title label in the ribbon bar.

4.3.4 RibbonTitleWidget

class ribbon.titlewidget.RibbonTitleWidget(title='PyQtRibbon', parent=None)

The title widget of the ribbon.

Methods

<code>addApplicationOptionAction(action)</code>	Add a display option to the category.
<code>addQuickAccessButton(button)</code>	Add a widget to the quick access bar.
<code>addRightToolButton(button)</code>	Add a widget to the right button bar.
<code>applicationButton()</code>	Return the application button.
<code>applicationMenu()</code>	Return the application menu.
<code>collapseRibbonButton()</code>	Return the collapse ribbon button.
<code>quickAccessButtons()</code>	Return the quick access buttons of the ribbon.
<code>quickAccessToolBar()</code>	Return the quick access toolbar of the ribbon.
<code>removeCollapseButton()</code>	Remove the min button from the ribbon.
<code>removeHelpButton()</code>	Remove the help button from the ribbon.
<code>rightToolBar()</code>	Return the right toolbar of the ribbon.
<code>setApplicationIcon(icon)</code>	Set the application icon.
<code>setCollapseButtonIcon(icon)</code>	Set the icon of the min button.
<code>setHelpButtonIcon(icon)</code>	Set the icon of the help button.
<code>setQuickAccessButtonHeight([height])</code>	Set the height of the quick access buttons.
<code>setRightToolBarHeight([height])</code>	Set the height of the right buttons.
<code>setTabBarHeight([height])</code>	Set the height of the tab bar.
<code>setTitle(title)</code>	Set the title of the ribbon.
<code>tabBar()</code>	Return the tab bar of the ribbon.
<code>tabBarHeight()</code>	Get the height of the tab bar.
<code>title()</code>	Return the title of the ribbon.

collapseRibbonButtonClicked	
helpButtonClicked	

addApplicationOptionAction(*action: QAction*)

Add a display option to the category.

Parameters

action – The action of the display option.

addQuickAccessButton(*button: QToolButton*)

Add a widget to the quick access bar.

Parameters

button – The button to add.

addRightToolButton(*button: QToolButton*)

Add a widget to the right button bar.

Parameters

button – The button to add.

applicationButton()

Return the application button.

applicationMenu() → QMenu

Return the application menu.

Returns

The application menu.

collapseRibbonButton() → QToolButton

Return the collapse ribbon button.

Returns

The collapse ribbon button.

helpButtonClicked

Signal: The help button was clicked.

quickAccessButtons() → list[PyQt5.QtWidgets.QToolButton]

Return the quick access buttons of the ribbon.

Returns

The quick access buttons of the ribbon.

quickAccessToolBar() → QToolBar

Return the quick access toolbar of the ribbon.

Returns

The quick access toolbar of the ribbon.

removeCollapseButton()

Remove the min button from the ribbon.

removeHelpButton()

Remove the help button from the ribbon.

rightToolBar() → QToolBar

Return the right toolbar of the ribbon.

Returns

The right toolbar of the ribbon.

setApplicationIcon(icon: QIcon)

Set the application icon.

Parameters

icon – The icon to set.

setCollapseButtonIcon(icon: QIcon)

Set the icon of the min button.

Parameters

icon – The icon to set.

setHelpButtonIcon(icon: QIcon)

Set the icon of the help button.

Parameters

icon – The icon to set.

setQuickAccessButtonHeight(height: int = 40)

Set the height of the quick access buttons.

Parameters

height – The height to set.

setRightToolBarHeight(height: int = 24)

Set the height of the right buttons.

Parameters

height – The height to set.

setTabBarHeight(height: int = 50)

Set the height of the tab bar.

Parameters

height – The height to set.

setTitle(title: str)

Set the title of the ribbon.

Parameters

title – The title to set.

tabBar() → *RibbonTabBar*

Return the tab bar of the ribbon.

Returns

The tab bar of the ribbon.

tabBarHeight() → int

Get the height of the tab bar.

Returns

The height of the tab bar.

title()

Return the title of the ribbon.

Returns

The title of the ribbon.

4.4 Ribbon Category

4.4.1 RibbonCategory

```
class ribbon.category.RibbonCategory(title: str, style: RibbonCategoryStyle =  
                                     RibbonCategoryStyle.Normal, color: Optional[QColor] = None,  
                                     parent=None)
```

The RibbonCategory is the logical grouping that represents the contents of a ribbon tab.

Methods

<code>addPanel(title)</code>	Add a new panel to the category.
<code>categoryStyle()</code>	Return the button style of the category.
<code>panel(title)</code>	Return a panel from the category.
<code>removePanel(title)</code>	Remove a panel from the category.
<code>setCategoryStyle(style)</code>	Set the button style of the category.
<code>takePanel(title)</code>	Remove and return a panel from the category.
<code>title()</code>	Return the title of the category.

displayOptionsButtonClicked	
------------------------------------	--

addPanel (*title: str*) → *RibbonPanel*

Add a new panel to the category.

Parameters**title** – The title of the panel.**Returns**

The newly created panel.

categoryStyle()

Return the button style of the category.

Returns

The button tyle.

displayOptionsButtonClicked

The signal that is emitted when the display options button is clicked.

panel (*title: str*)

Return a panel from the category.

Parameters**title** – The title of the panel.

Returns

The panel.

removePanel(*title: str*)

Remove a panel from the category.

Parameters

title – The title of the panel.

setCategoryStyle(*style: RibbonCategoryStyle*)

Set the button style of the category.

Parameters

style – The button style.

takePanel(*title: str*)

Remove and return a panel from the category.

Parameters

title – The title of the panel.

Returns

The removed panel.

title() → str

Return the title of the category.

4.4.2 RibbonNormalCategory

class ribbon.category.**RibbonNormalCategory**(*title: str, parent: QWidget*)

A normal category.

Methods

setCategoryStyle (<i>style</i>)	Set the button style of the category.
--	---------------------------------------

setCategoryStyle(*style: RibbonCategoryStyle*)

Set the button style of the category.

Parameters

style – The button style.

4.4.3 RibbonContextCategory

class ribbon.category.**RibbonContextCategory**(*title: str, color: QColor, parent: QWidget*)

A context category.

Methods

<code>categoryVisible()</code>	Return whether the category is shown.
<code>color()</code>	Return the color of the context category.
<code>hideContextCategory()</code>	Hide the given category, if it is not a context category, nothing happens.
<code>setCategoryStyle(style)</code>	Set the button style of the category.
<code>setCategoryVisible(state)</code>	Set the state of the category.
<code>setColor(color)</code>	Set the color of the context category.
<code>showContextCategory()</code>	Show the given category, if it is not a context category, nothing happens.

categoryVisible() → bool

Return whether the category is shown.

Returns

Whether the category is shown.

color() → QColor

Return the color of the context category.

Returns

The color of the context category.

hideContextCategory()

Hide the given category, if it is not a context category, nothing happens.

setCategoryStyle(*style*: [RibbonCategoryStyle](#))

Set the button style of the category.

Parameters

style – The button style.

setCategoryVisible(*state*: bool)

Set the state of the category.

Parameters

state – The state.

setColor(*color*: QColor)

Set the color of the context category.

Parameters

color – The color of the context category.

showContextCategory()

Show the given category, if it is not a context category, nothing happens.

4.4.4 RibbonCategoryStyle

class ribbon.category.RibbonCategoryStyle(*value*)

The button style of a category.

4.4.5 RibbonCategoryScrollArea

class ribbon.categorylayoutwidget.RibbonCategoryScrollArea

Scroll area for the gallery

4.4.6 RibbonCategoryScrollAreaContents

class ribbon.categorylayoutwidget.RibbonCategoryScrollAreaContents

Scroll area contents for the gallery

4.4.7 RibbonCategoryLayoutButton

class ribbon.categorylayoutwidget.RibbonCategoryLayoutButton

Previous/Next buttons in the category when the size is not enough for the widgets.

4.4.8 RibbonCategoryLayoutWidget

class ribbon.categorylayoutwidget.RibbonCategoryLayoutWidget(*parent=None*)

The category layout widget's category scroll area to arrange the widgets in the category.

Methods

<i>addWidget(widget)</i>	Add a widget to the category layout.
<i>autoSetScrollButtonsVisible()</i>	Set the visibility of the scroll buttons.
<i>paintEvent(a0)</i>	Override the paint event to draw the background.
<i>removeWidget(widget)</i>	Remove a widget from the category layout.
<i>resizeEvent(a0)</i>	Override the resize event to resize the scroll area.
<i>scrollNext()</i>	Scroll the category to the next widget.
<i>scrollPrevious()</i>	Scroll the category to the previous widget.
<i>takeWidget(widget)</i>	Remove and return a widget from the category layout.

displayOptionsButtonClicked	
-----------------------------	--

addWidget(*widget: QWidget*)

Add a widget to the category layout.

Parameters

widget – The widget to add.

autoSetScrollButtonsVisible()

Set the visibility of the scroll buttons.

paintEvent(*a0: QPaintEvent*) → None

Override the paint event to draw the background.

removeWidget(*widget: QWidget*)

Remove a widget from the category layout.

Parameters

widget – The widget to remove.

resizeEvent(*a0: QResizeEvent*) → None

Override the resize event to resize the scroll area.

scrollNext()

Scroll the category to the next widget.

scrollPrevious()

Scroll the category to the previous widget.

takeWidget(*widget: QWidget*)

Remove and return a widget from the category layout.

Parameters

widget – The widget to remove.

Returns

The widget that was removed.

4.5 Ribbon Panel

4.5.1 RibbonPanel

class ribbon.panel.**RibbonPanel**(*title: str, maxRows=6, parent=None*)

Panel in the ribbon category.

Methods

<code>addButton([text, icon, style, showText, ...])</code>	Add a button to the panel.
<code>addCalendarWidget([rowSpan, colSpan, mode, ...])</code>	Add a calendar widget to the panel.
<code>addComboBox(items[, rowSpan, colSpan, mode, ...])</code>	Add a combo box to the panel.
<code>addDateEdit([rowSpan, colSpan, mode, alignment])</code>	Add a date edit to the panel.
<code>addDateTimeEdit([rowSpan, colSpan, mode, ...])</code>	Add a date time edit to the panel.
<code>addDoubleSpinBox([rowSpan, colSpan, mode, ...])</code>	Add a double spin box to the panel.
<code>addFontComboBox([rowSpan, colSpan, mode, ...])</code>	Add a font combo box to the panel.
<code>addGallery([minimumWidth, popupHideOnClick, ...])</code>	Add a gallery to the panel.
<code>addHorizontalSeparator([linewidth, rowSpan, ...])</code>	Add a horizontal separator to the panel.
<code>addLabel(text[, rowSpan, colSpan, mode, ...])</code>	Add a label to the panel.
<code>addLargeButton([text, icon, showText, ...])</code>	Add a large button to the panel.

continues on next page

Table 2 – continued from previous page

<code>addLargeToggleButton</code> ([text, icon, showText, ...])	Add a large toggle button to the panel.
<code>addLargeWidget</code> (widget[, mode, alignment])	Add a large widget to the panel.
<code>addLineEdit</code> ([rowSpan, colSpan, mode, alignment])	Add a line edit to the panel.
<code>addListWidget</code> ([rowSpan, colSpan, mode, ...])	Add a list widget to the panel.
<code>addMediumButton</code> ([text, icon, showText, ...])	Add a medium button to the panel.
<code>addMediumToggleButton</code> ([text, icon, ...])	Add a medium toggle button to the panel.
<code>addMediumWidget</code> (widget[, mode, alignment])	Add a medium widget to the panel.
<code>addPlainTextEdit</code> ([rowSpan, colSpan, mode, ...])	Add a plain text edit to the panel.
<code>addProgressBar</code> ([rowSpan, colSpan, mode, ...])	Add a progress bar to the panel.
<code>addSeparator</code> ([orientation, width, rowSpan, ...])	Add a separator to the panel.
<code>addSlider</code> ([rowSpan, colSpan, mode, alignment])	Add a slider to the panel.
<code>addSmallButton</code> ([text, icon, showText, ...])	Add a small button to the panel.
<code>addSmallToggleButton</code> ([text, icon, showText, ...])	Add a small toggle button to the panel.
<code>addSmallWidget</code> (widget[, mode, alignment])	Add a small widget to the panel.
<code>addSpinBox</code> ([rowSpan, colSpan, mode, alignment])	Add a spin box to the panel.
<code>addTableWidget</code> ([rowSpan, colSpan, mode, ...])	Add a table widget to the panel.
<code>addTextEdit</code> ([rowSpan, colSpan, mode, alignment])	Add a text edit to the panel.
<code>addTimeEdit</code> ([rowSpan, colSpan, mode, alignment])	Add a time edit to the panel.
<code>addToggleButton</code> ([text, icon, style, ...])	Add a toggle button to the panel.
<code>addTreeWidget</code> ([rowSpan, colSpan, mode, ...])	Add a tree widget to the panel.
<code>addVerticalSeparator</code> ([linewidth, rowSpan, ...])	Add a vertical separator to the panel.
<code>addWidget</code> (widget[, rowSpan, colSpan, mode, ...])	Add a widget to the panel.
<code>panelOptionButton</code> ()	Return the panel option button.
<code>removeWidget</code> (widget)	Remove a widget from the panel.
<code>rowHeight</code> ()	Return the height of a row.
<code>setPanelOptionToolTip</code> (text)	Set the tooltip of the panel option button.
<code>setTitleText</code> (text)	Set the title text of the panel.
<code>titleText</code> ()	Get the title text of the panel.
<code>widget</code> (index)	Get the widget at the given index.

panelOptionClicked	
---------------------------	--

addButton(text: Optional[str] = None, icon: Optional[QIcon] = None, style: RibbonButtonStyle = RibbonButtonStyle.Large, showText: bool = True, colSpan: int = 1, slot=None, shortcut=None, tooltip=None, statusTip=None, mode=RibbonSpaceFindMode.ColumnWise, alignment=132) → RibbonToolButton

Add a button to the panel.

Parameters

- **text** – The text of the button.
- **icon** – The icon of the button.
- **style** – The style of the button.
- **showText** – Whether to show the text of the button.
- **colSpan** – The number of columns the button should span.
- **slot** – The slot to call when the button is clicked.

- **shortcut** – The shortcut of the button.
- **tooltip** – The tooltip of the button.
- **statusTip** – The status tip of the button.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the button.

Returns

The button that was added.

addCalendarWidget(*rowSpan: int = 6, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → QCalendarWidget

Add a calendar widget to the panel.

Parameters

- **rowSpan** – The number of rows the calendar widget should span.
- **colSpan** – The number of columns the calendar widget should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the calendar widget.

addComboBox(*items: List[str], rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → QComboBox

Add a combo box to the panel.

Parameters

- **items** – The items of the combo box.
- **rowSpan** – The number of rows the combo box should span.
- **colSpan** – The number of columns the combo box should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the combo box.

addDateEdit(*rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → QDateEdit

Add a date edit to the panel.

Parameters

- **rowSpan** – The number of rows the date edit should span.
- **colSpan** – The number of columns the date edit should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the date edit.

addDateTimeEdit(*rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → QDateTimeEdit

Add a date time edit to the panel.

Parameters

- **rowSpan** – The number of rows the date time edit should span.
- **colSpan** – The number of columns the date time edit should span.

- **mode** – The mode to find spaces.
- **alignment** – The alignment of the date time edit.

addDoubleSpinBox(*rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → *QDoubleSpinBox*

Add a double spin box to the panel.

Parameters

- **rowSpan** – The number of rows the double spin box should span.
- **colSpan** – The number of columns the double spin box should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the double spin box.

addFontComboBox(*rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → *QFontComboBox*

Add a font combo box to the panel.

Parameters

- **rowSpan** – The number of rows the combo box should span.
- **colSpan** – The number of columns the combo box should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the combo box.

addGallery(*minimumWidth=800, popupHideOnClick=False, rowSpan: int = 6, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise*) → *RibbonGallery*

Add a gallery to the panel.

Parameters

- **minimumWidth** – The minimum width of the gallery.
- **popupHideOnClick** – Whether the gallery popup should be hidden when a user clicks on it.
- **rowSpan** – The number of rows the gallery spans.
- **colSpan** – The number of columns the gallery spans.
- **mode** – The mode of the gallery.

Returns

The gallery.

addHorizontalSeparator(*linewidth=6, rowSpan: int = 1, colSpan: int = 2, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → *RibbonHorizontalSeparator*

Add a horizontal separator to the panel.

Parameters

- **linewidth** – The width of the separator.
- **rowSpan** – The number of rows the separator spans.
- **colSpan** – The number of columns the separator spans.
- **mode** – The mode to find spaces.

- **alignment** – The alignment of the separator.

Returns

The separator.

addLabel(*text: str, rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → `QLabel`

Add a label to the panel.

Parameters

- **text** – The text of the label.
- **rowSpan** – The number of rows the label should span.
- **colSpan** – The number of columns the label should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the label.

addLargeButton(*text: Optional[str] = None, icon: Optional[QIcon] = None, showText: bool = True, colSpan: int = 1, slot=None, shortcut=None, tooltip=None, statusTip=None, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → `RibbonToolButton`

Add a large button to the panel.

Parameters

- **text** – The text of the button.
- **icon** – The icon of the button.
- **showText** – Whether to show the text of the button.
- **colSpan** – The number of columns the button should span.
- **slot** – The slot to call when the button is clicked.
- **shortcut** – The shortcut of the button.
- **tooltip** – The tooltip of the button.
- **statusTip** – The status tip of the button.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the button.

addLargeToggleButton(*text: Optional[str] = None, icon: Optional[QIcon] = None, showText: bool = True, colSpan: int = 1, slot=None, shortcut=None, tooltip=None, statusTip=None, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → `RibbonToolButton`

Add a large toggle button to the panel.

Parameters

- **text** – The text of the button.
- **icon** – The icon of the button.
- **showText** – Whether to show the text of the button.
- **colSpan** – The number of columns the button should span.
- **slot** – The slot to call when the button is clicked.
- **shortcut** – The shortcut of the button.

- **tooltip** – The tooltip of the button.
- **statusTip** – The status tip of the button.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the button.

addLargeWidget(*widget: QWidget, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*)

Add a large widget to the panel.

Parameters

- **widget** – The widget to add.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the widget.

addLineEdit(*rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → *QLineEdit*

Add a line edit to the panel.

Parameters

- **rowSpan** – The number of rows the line edit should span.
- **colSpan** – The number of columns the line edit should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the line edit.

addListWidget(*rowSpan: int = 6, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → *QListWidget*

Add a list widget to the panel.

Parameters

- **rowSpan** – The number of rows the list widget should span.
- **colSpan** – The number of columns the list widget should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the list widget.

addMediumButton(*text: Optional[str] = None, icon: Optional[QIcon] = None, showText: bool = True, colSpan: int = 1, slot=None, shortcut=None, tooltip=None, statusTip=None, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → *RibbonToolButton*

Add a medium button to the panel.

Parameters

- **text** – The text of the button.
- **icon** – The icon of the button.
- **showText** – Whether to show the text of the button.
- **colSpan** – The number of columns the button should span.
- **slot** – The slot to call when the button is clicked.
- **shortcut** – The shortcut of the button.
- **tooltip** – The tooltip of the button.

- **statusTip** – The status tip of the button.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the button.

addMediumToggleButton(*text: Optional[str] = None, icon: Optional[QIcon] = None, showText: bool = True, colSpan: int = 1, slot=None, shortcut=None, tooltip=None, statusTip=None, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → *RibbonToolButton*

Add a medium toggle button to the panel.

Parameters

- **text** – The text of the button.
- **icon** – The icon of the button.
- **showText** – Whether to show the text of the button.
- **colSpan** – The number of columns the button should span.
- **slot** – The slot to call when the button is clicked.
- **shortcut** – The shortcut of the button.
- **tooltip** – The tooltip of the button.
- **statusTip** – The status tip of the button.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the button.

addMediumWidget(*widget: QWidget, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*)

Add a medium widget to the panel.

Parameters

- **widget** – The widget to add.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the widget.

addPlainTextEdit(*rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → *QPlainTextEdit*

Add a plain text edit to the panel.

Parameters

- **rowSpan** – The number of rows the text edit should span.
- **colSpan** – The number of columns the text edit should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the text edit.

addProgressBar(*rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → *QProgressBar*

Add a progress bar to the panel.

Parameters

- **rowSpan** – The number of rows the progress bar should span.

- **colSpan** – The number of columns the progress bar should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the progress bar.

addSeparator(*orientation=2, width=6, rowSpan: int = 6, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → Union[[RibbonHorizontalSeparator](#), [RibbonVerticalSeparator](#)]

Add a separator to the panel.

Parameters

- **orientation** – The orientation of the separator.
- **width** – The width of the separator.
- **rowSpan** – The number of rows the separator spans.
- **colSpan** – The number of columns the separator spans.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the separator.

Returns

The separator.

addSlider(*rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → [QSlider](#)

Add a slider to the panel.

Parameters

- **rowSpan** – The number of rows the slider should span.
- **colSpan** – The number of columns the slider should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the slider.

addSmallButton(*text: Optional[str] = None, icon: Optional[QIcon] = None, showText: bool = True, colSpan: int = 1, slot=None, shortcut=None, tooltip=None, statusTip=None, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → [RibbonToolButton](#)

Add a small button to the panel.

Parameters

- **text** – The text of the button.
- **icon** – The icon of the button.
- **showText** – Whether to show the text of the button.
- **colSpan** – The number of columns the button should span.
- **slot** – The slot to call when the button is clicked.
- **shortcut** – The shortcut of the button.
- **tooltip** – The tooltip of the button.
- **statusTip** – The status tip of the button.
- **mode** – The mode to find spaces.

- **alignment** – The alignment of the button.

addSmallToggleButton(*text: Optional[str] = None, icon: Optional[QIcon] = None, showText: bool = True, colSpan: int = 1, slot=None, shortcut=None, tooltip=None, statusTip=None, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → *RibbonToolButton*

Add a small toggle button to the panel.

Parameters

- **text** – The text of the button.
- **icon** – The icon of the button.
- **showText** – Whether to show the text of the button.
- **colSpan** – The number of columns the button should span.
- **slot** – The slot to call when the button is clicked.
- **shortcut** – The shortcut of the button.
- **tooltip** – The tooltip of the button.
- **statusTip** – The status tip of the button.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the button.

addSmallWidget(*widget: QWidget, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*)

Add a small widget to the panel.

Parameters

- **widget** – The widget to add.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the widget.

Returns

The widget that was added.

addSpinBox(*rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → *QSpinBox*

Add a spin box to the panel.

Parameters

- **rowSpan** – The number of rows the spin box should span.
- **colSpan** – The number of columns the spin box should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the spin box.

addTableWidget(*rowSpan: int = 6, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → *QTableWidget*

Add a table widget to the panel.

Parameters

- **rowSpan** – The number of rows the table widget should span.
- **colSpan** – The number of columns the table widget should span.

- **mode** – The mode to find spaces.
- **alignment** – The alignment of the table widget.

addTextEdit(*rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → QTextEdit

Add a text edit to the panel.

Parameters

- **rowSpan** – The number of rows the text edit should span.
- **colSpan** – The number of columns the text edit should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the text edit.

addTimeEdit(*rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → QTimeEdit

Add a time edit to the panel.

Parameters

- **rowSpan** – The number of rows the time edit should span.
- **colSpan** – The number of columns the time edit should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the time edit.

addToggleButton(*text: Optional[str] = None, icon: Optional[QIcon] = None, style: RibbonButtonStyle = RibbonButtonStyle.Large, showText: bool = True, colSpan: int = 1, slot=None, shortcut=None, tooltip=None, statusTip=None, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → RibbonToolButton

Add a toggle button to the panel.

Parameters

- **text** – The text of the button.
- **icon** – The icon of the button.
- **style** – The style of the button.
- **showText** – Whether to show the text of the button.
- **colSpan** – The number of columns the button should span.
- **slot** – The slot to call when the button is clicked.
- **shortcut** – The shortcut of the button.
- **tooltip** – The tooltip of the button.
- **statusTip** – The status tip of the button.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the button.

addTreeWidget(*rowSpan: int = 6, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → QTreeWidget

Add a tree widget to the panel.

Parameters

- **rowSpan** – The number of rows the tree widget should span.
- **colSpan** – The number of columns the tree widget should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the tree widget.

addVerticalSeparator(*linewidth=6, rowSpan: int = 6, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*) → *RibbonVerticalSeparator*

Add a vertical separator to the panel.

Parameters

- **linewidth** – The width of the separator.
- **rowSpan** – The number of rows the separator spans.
- **colSpan** – The number of columns the separator spans.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the separator.

Returns

The separator.

addWidget(*widget: QWidget, rowSpan: int = 2, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise, alignment=132*)

Add a widget to the panel.

Parameters

- **widget** – The widget to add.
- **rowSpan** – The number of rows the widget should span, 2: small, 3: medium, 6: large.
- **colSpan** – The number of columns the widget should span.
- **mode** – The mode to find spaces.
- **alignment** – The alignment of the widget.

panelOptionButton() → *RibbonPanelOptionButton*

Return the panel option button.

Returns

The panel option button.

removeWidget(*widget: QWidget*)

Remove a widget from the panel.

rowHeight() → int

Return the height of a row.

setPanelOptionToolTip(*text: str*)

Set the tooltip of the panel option button.

Parameters

- **text** – The tooltip text.

setTitleText(*text: str*)

Set the title text of the panel.

Parameters

text – The text to set.

titleText()

Get the title text of the panel.

Returns

The title text.

widget(*index: int*) → QWidget

Get the widget at the given index.

Parameters

index – The index of the widget, starting from 0.

Returns

The widget at the given index.

4.5.2 RibbonPanellItemWidget

class ribbon.panel.**RibbonPanellItemWidget**(*parent=None*)

Widget to display a panel item.

Methods

<i>addWidget</i>(widget)	Add a widget to the panel item.
--	---------------------------------

addWidget(*widget*)

Add a widget to the panel item.

Parameters

widget – The widget to add.

4.5.3 RibbonSpaceFindMode

class ribbon.panel.**RibbonSpaceFindMode**(*value*)

Mode to find available space in a grid layout, ColumnWise or RowWise.

4.5.4 RibbonGridLayoutManager

class ribbon.panel.**RibbonGridLayoutManager**(*rows: int*)

Grid Layout Manager.

Methods

<code>request_cells([rowSpan, colSpan, mode])</code>	Request a number of available cells from the grid.
--	--

request_cells(*rowSpan: int = 1, colSpan: int = 1, mode=RibbonSpaceFindMode.ColumnWise*)

Request a number of available cells from the grid.

Parameters

- **rowSpan** – The number of rows the cell should span.
- **colSpan** – The number of columns the cell should span.
- **mode** – The mode of the grid.

Returns

row, col, the row and column of the requested cell.

4.6 Ribbon Gallery

4.6.1 RibbonGallery

class ribbon.gallery.**RibbonGallery**(*minimumWidth=800, popupHideOnClick=False, parent=None*)

A widget that displays a gallery of buttons.

Methods

<code>addButton([text, icon, slot, shortcut, ...])</code>	Add a button to the gallery
<code>addToggleButton([text, icon, slot, ...])</code>	Add a toggle button to the gallery
<code>popupWindowSize()</code>	Return the size of the popup window
<code>resizeEvent(a0)</code>	Resize the gallery.
<code>setPopupHideOnClick(popupHideOnClick)</code>	Set the hide on click flag
<code>setPopupWindowSize(size)</code>	Set the size of the popup window
<code>setSelectedButton()</code>	Set the selected button
<code>showPopup()</code>	Show the popup window

addButton(*text: Optional[str] = None, icon: Optional[QIcon] = None, slot=None, shortcut=None, tooltip=None, statusTip=None, checkable=False*) → [RibbonToolButton](#)

Add a button to the gallery

Parameters

- **text** – text of the button
- **icon** – icon of the button
- **slot** – slot to call when the button is clicked
- **shortcut** – shortcut of the button
- **tooltip** – tooltip of the button
- **statusTip** – status tip of the button
- **checkable** – checkable flag of the button

Returns

the button added

addToggleButton(*text: Optional[str] = None, icon: Optional[QIcon] = None, slot=None, shortcut=None, tooltip=None, statusTip=None*) → *RibbonToolButton*

Add a toggle button to the gallery

Parameters

- **text** – text of the button
- **icon** – icon of the button
- **slot** – slot to call when the button is clicked
- **shortcut** – shortcut of the button
- **tooltip** – tooltip of the button
- **statusTip** – status tip of the button

Returns

the button added

popupWindowSize()

Return the size of the popup window

Returns

size of the popup window

resizeEvent(*a0: QResizeEvent*) → None

Resize the gallery.

setPopupHideOnClick(*popupHideOnClick: bool*)

Set the hide on click flag

Parameters

popupHideOnClick – hide on click flag

setPopupWindowSize(*size: QSize*)

Set the size of the popup window

Parameters

size – size of the popup window

setSelectedButton()

Set the selected button

showPopup()

Show the popup window

4.6.2 RibbonGalleryListWidget

class ribbon.gallery.**RibbonGalleryListWidget**(*parent=None*)

Gallery list widget.

Methods

<code>resizeEvent(e)</code>	Resize the list widget.
<code>scrollToNextRow()</code>	Scroll to the next row.
<code>scrollToPreviousRow()</code>	Scroll to the previous row.

resizeEvent(*e: QResizeEvent*) → None

Resize the list widget.

scrollToNextRow() → None

Scroll to the next row.

scrollToPreviousRow() → None

Scroll to the previous row.

4.6.3 RibbonGalleryButton

class ribbon.gallery.RibbonGalleryButton

Gallery button.

4.6.4 RibbonGalleryPopupListWidget

class ribbon.gallery.RibbonGalleryPopupListWidget(*parent=None*)

Gallery popup list widget.

4.7 Ribbon Tool Button

4.7.1 RibbonToolButton

class ribbon.toolbutton.RibbonToolButton(*parent=None*)

Tool button that is showed in the ribbon.

Methods

<code>addAction(action)</code>	Add an action to the button.
<code>addActions(actions)</code>	Add actions to the button.
<code>setButtonStyle(style)</code>	Set the button style of the button.
<code>setMenu(menu)</code>	Set the menu of the button.

addAction(*action: QAction*) → None

Add an action to the button.

Parameters

action – The action to add.

addAction(*actions: Iterable[QAction]*) → None

Add actions to the button.

Parameters

actions – The actions to add.

setButtonStyle(*style: RibbonButtonStyle*)

Set the button style of the button.

Parameters

style – The button style of the button.

setMenu(*menu: QMenu*)

Set the menu of the button.

Parameters

menu – The menu of the button.

4.7.2 RibbonButtonStyle

class ribbon.toolbutton.RibbonButtonStyle(*value*)

Button style, Small, Medium, or Large.

4.8 Ribbon Separator

4.8.1 RibbonSeparator

class ribbon.separator.RibbonSeparator(*orientation=2, width=6, parent=None*)

The RibbonSeparator is a separator that can be used to separate widgets in a ribbon.

Methods

<i>paintEvent</i> (<i>event</i>)	Paint the separator.
<i>setTopBottomMargins</i> (<i>top, bottom</i>)	Set the top and bottom margins.
<i>sizeHint</i> ()	Return the size hint.

paintEvent(*event: QPaintEvent*) → None

Paint the separator.

setTopBottomMargins(*top: int, bottom: int*) → None

Set the top and bottom margins.

sizeHint() → QSize

Return the size hint.

4.8.2 RibbonHorizontalSeparator

```
class ribbon.separator.RibbonHorizontalSeparator(width: int = 6, parent=None)
    Horizontal separator.
```

4.8.3 RibbonVerticalSeparator

```
class ribbon.separator.RibbonVerticalSeparator(width: int = 6, parent=None)
    Vertical separator.
```

INDICES AND TABLES

- genindex
- modindex
- search

A

- `addAction()` (*ribbon.toolbar.RibbonToolBar method*), 38
- `addActions()` (*ribbon.toolbar.RibbonToolBar method*), 38
- `addApplicationOptionAction()` (*ribbon.ribbonbar.RibbonBar method*), 12
- `addApplicationOptionAction()` (*ribbon.titlewidget.RibbonTitleWidget method*), 18
- `addButton()` (*ribbon.gallery.RibbonGallery method*), 36
- `addButton()` (*ribbon.panel.RibbonPanel method*), 25
- `addCalendarWidget()` (*ribbon.panel.RibbonPanel method*), 26
- `addCategory()` (*ribbon.ribbonbar.RibbonBar method*), 12
- `addComboBox()` (*ribbon.panel.RibbonPanel method*), 26
- `addContextCategory()` (*ribbon.ribbonbar.RibbonBar method*), 12
- `addDateEdit()` (*ribbon.panel.RibbonPanel method*), 26
- `addDateTimeEdit()` (*ribbon.panel.RibbonPanel method*), 26
- `addDoubleSpinBox()` (*ribbon.panel.RibbonPanel method*), 27
- `addFontComboBox()` (*ribbon.panel.RibbonPanel method*), 27
- `addGallery()` (*ribbon.panel.RibbonPanel method*), 27
- `addHorizontalSeparator()` (*ribbon.panel.RibbonPanel method*), 27
- `addLabel()` (*ribbon.panel.RibbonPanel method*), 28
- `addLargeButton()` (*ribbon.panel.RibbonPanel method*), 28
- `addLargeToggleButton()` (*ribbon.panel.RibbonPanel method*), 28
- `addLargeWidget()` (*ribbon.panel.RibbonPanel method*), 29
- `addLineEdit()` (*ribbon.panel.RibbonPanel method*), 29
- `addListWidget()` (*ribbon.panel.RibbonPanel method*), 29
- `addMediumButton()` (*ribbon.panel.RibbonPanel method*), 29
- `addMediumToggleButton()` (*ribbon.panel.RibbonPanel method*), 30
- `addMediumWidget()` (*ribbon.panel.RibbonPanel method*), 30
- `addNormalCategory()` (*ribbon.ribbonbar.RibbonBar method*), 13
- `addPanel()` (*ribbon.category.RibbonCategory method*), 20
- `addPlainTextEdit()` (*ribbon.panel.RibbonPanel method*), 30
- `addProgressBar()` (*ribbon.panel.RibbonPanel method*), 30
- `addQuickAccessButton()` (*ribbon.ribbonbar.RibbonBar method*), 13
- `addQuickAccessButton()` (*ribbon.titlewidget.RibbonTitleWidget method*), 18
- `addRightToolBar()` (*ribbon.ribbonbar.RibbonBar method*), 13
- `addRightToolBar()` (*ribbon.titlewidget.RibbonTitleWidget method*), 18
- `addSeparator()` (*ribbon.panel.RibbonPanel method*), 31
- `addSlider()` (*ribbon.panel.RibbonPanel method*), 31
- `addSmallButton()` (*ribbon.panel.RibbonPanel method*), 31
- `addSmallToggleButton()` (*ribbon.panel.RibbonPanel method*), 32
- `addSmallWidget()` (*ribbon.panel.RibbonPanel method*), 32
- `addSpinBox()` (*ribbon.panel.RibbonPanel method*), 32
- `addTab()` (*ribbon.tabbar.RibbonTabBar method*), 16
- `addTableWidget()` (*ribbon.panel.RibbonPanel method*), 32
- `addTextEdit()` (*ribbon.panel.RibbonPanel method*), 33
- `addTimeEdit()` (*ribbon.panel.RibbonPanel method*), 33
- `addToggleButton()` (*ribbon.gallery.RibbonGallery method*), 37
- `addToggleButton()` (*ribbon.panel.RibbonPanel method*), 33
- `addTreeWidget()` (*ribbon.panel.RibbonPanel method*), 33

33
addVerticalSeparator() (ribbon.panel.RibbonPanel method), 34
addWidget() (ribbon.categorylayoutwidget.RibbonCategoryLayoutWidget method), 23
addWidget() (ribbon.panel.RibbonPanel method), 34
addWidget() (ribbon.panel.RibbonPanelItemWidget method), 35
applicationButton() (ribbon.titlewidget.RibbonTitleWidget method), 18
applicationMenu() (ribbon.titlewidget.RibbonTitleWidget method), 18
applicationOptionButton() (ribbon.ribbonbar.RibbonBar method), 13
autoSetScrollButtonsVisible() (ribbon.categorylayoutwidget.RibbonCategoryLayoutWidget method), 23

C
categories() (ribbon.ribbonbar.RibbonBar method), 13
categoryStyle() (ribbon.category.RibbonCategory method), 20
categoryVisible() (ribbon.category.RibbonContextCategory method), 22
collapseRibbonButton() (ribbon.ribbonbar.RibbonBar method), 13
collapseRibbonButton() (ribbon.titlewidget.RibbonTitleWidget method), 18
color() (ribbon.category.RibbonContextCategory method), 22
currentTabColor() (ribbon.tabbar.RibbonTabBar method), 16

D
displayOptionsButtonClicked (ribbon.category.RibbonCategory attribute), 20

H
helpButtonClicked (ribbon.ribbonbar.RibbonBar attribute), 13
helpButtonClicked (ribbon.titlewidget.RibbonTitleWidget attribute), 18
hideContextCategory() (ribbon.category.RibbonContextCategory method), 22
hideContextCategory() (ribbon.ribbonbar.RibbonBar method), 13
hideRibbon() (ribbon.ribbonbar.RibbonBar method), 13

I
indexOf() (ribbon.tabbar.RibbonTabBar method), 17

L
layout() (ribbon.mainwindow.RibbonMainWindow method), 11

M
minimumSizeHint() (ribbon.ribbonbar.RibbonBar method), 13

P
paintEvent() (ribbon.categorylayoutwidget.RibbonCategoryLayoutWidget method), 23
paintEvent() (ribbon.separator.RibbonSeparator method), 39
paintEvent() (ribbon.tabbar.RibbonTabBar method), 17
panel() (ribbon.category.RibbonCategory method), 20
panelOptionButton() (ribbon.panel.RibbonPanel method), 34
popupWindowSize() (ribbon.gallery.RibbonGallery method), 37

Q
quickAccessButtons() (ribbon.titlewidget.RibbonTitleWidget method), 18
quickAccessToolBar() (ribbon.ribbonbar.RibbonBar method), 14
quickAccessToolBar() (ribbon.titlewidget.RibbonTitleWidget method), 18

R
removeCategory() (ribbon.ribbonbar.RibbonBar method), 14
removeCollapseButton() (ribbon.ribbonbar.RibbonBar method), 14
removeCollapseButton() (ribbon.titlewidget.RibbonTitleWidget method), 18
removeHelpButton() (ribbon.ribbonbar.RibbonBar method), 14
removeHelpButton() (ribbon.titlewidget.RibbonTitleWidget method), 18
removePanel() (ribbon.category.RibbonCategory method), 21

removeWidget() (ribbon.categorylayoutwidget.RibbonCategoryLayoutWidget method), 24
 removeWidget() (ribbon.panel.RibbonPanel method), 34
 request_cells() (ribbon.panel.RibbonGridLayoutManager method), 36
 resizeEvent() (ribbon.categorylayoutwidget.RibbonCategoryLayoutWidget method), 24
 resizeEvent() (ribbon.gallery.RibbonGallery method), 37
 resizeEvent() (ribbon.gallery.RibbonGalleryListWidget method), 38
 RibbonApplicationButton (class in ribbon.titlewidget), 16
 RibbonBar (class in ribbon.ribbonbar), 11
 ribbonBar() (ribbon.mainwindow.RibbonMainWindow method), 11
 RibbonButtonStyle (class in ribbon.toolbar), 39
 RibbonCategory (class in ribbon.category), 20
 RibbonCategoryLayoutButton (class in ribbon.categorylayoutwidget), 23
 RibbonCategoryLayoutWidget (class in ribbon.categorylayoutwidget), 23
 RibbonCategoryScrollArea (class in ribbon.categorylayoutwidget), 23
 RibbonCategoryScrollAreaContents (class in ribbon.categorylayoutwidget), 23
 RibbonCategoryStyle (class in ribbon.category), 23
 RibbonContextCategory (class in ribbon.category), 21
 RibbonGallery (class in ribbon.gallery), 36
 RibbonGalleryButton (class in ribbon.gallery), 38
 RibbonGalleryListWidget (class in ribbon.gallery), 37
 RibbonGalleryPopupListWidget (class in ribbon.gallery), 38
 RibbonGridLayoutManager (class in ribbon.panel), 35
 ribbonHeight() (ribbon.ribbonbar.RibbonBar method), 14
 RibbonHorizontalSeparator (class in ribbon.separator), 40
 RibbonMainWindow (class in ribbon.mainwindow), 11
 RibbonNormalCategory (class in ribbon.category), 21
 RibbonPanel (class in ribbon.panel), 24
 RibbonPanelItemWidget (class in ribbon.panel), 35
 RibbonSeparator (class in ribbon.separator), 39
 RibbonSpaceFindMode (class in ribbon.panel), 35
 RibbonTabBar (class in ribbon.tabbar), 16
 RibbonTitleLabel (class in ribbon.titlewidget), 17
 RibbonTitleWidget (class in ribbon.titlewidget), 17
 RibbonToolButton (class in ribbon.toolbar), 38
 RibbonVerticalSeparator (class in ribbon.separator), 40
 ribbonVisible() (ribbon.ribbonbar.RibbonBar method), 14
 rightToolBar() (ribbon.ribbonbar.RibbonBar method), 14
 rightToolBar() (ribbon.titlewidget.RibbonTitleWidget method), 18
 rowHeight() (ribbon.panel.RibbonPanel method), 34
 ScrollLayoutWidget
 scrollNext() (ribbon.categorylayoutwidget.RibbonCategoryLayoutWidget method), 24
 scrollPrevious() (ribbon.categorylayoutwidget.RibbonCategoryLayoutWidget method), 24
 scrollToNextRow() (ribbon.gallery.RibbonGalleryListWidget method), 38
 scrollToPreviousRow() (ribbon.gallery.RibbonGalleryListWidget method), 38
 setApplicationIcon() (ribbon.ribbonbar.RibbonBar method), 14
 setApplicationIcon() (ribbon.titlewidget.RibbonTitleWidget method), 19
 setButtonStyle() (ribbon.toolbar.RibbonToolButton method), 39
 setCategoryStyle() (ribbon.category.RibbonCategory method), 21
 setCategoryStyle() (ribbon.category.RibbonContextCategory method), 22
 setCategoryStyle() (ribbon.category.RibbonNormalCategory method), 21
 setCategoryVisible() (ribbon.category.RibbonContextCategory method), 22
 setCollapseButtonIcon() (ribbon.ribbonbar.RibbonBar method), 14
 setCollapseButtonIcon() (ribbon.titlewidget.RibbonTitleWidget method), 19
 setColor() (ribbon.category.RibbonContextCategory method), 22
 setCurrentCategory() (ribbon.ribbonbar.RibbonBar method), 14
 setHelpButtonIcon() (ribbon.ribbonbar.RibbonBar method), 14
 setHelpButtonIcon() (ribbon.titlewidget.RibbonTitleWidget method), 19

`setMenu()` (*ribbon.toolbutton.RibbonToolButton method*), 39

`setPanelOptionToolTip()` (*ribbon.panel.RibbonPanel method*), 34

`setPopupHideOnClick()` (*ribbon.gallery.RibbonGallery method*), 37

`setPopupWindowSize()` (*ribbon.gallery.RibbonGallery method*), 37

`setQuickAccessButtonHeight()` (*ribbon.ribbonbar.RibbonBar method*), 15

`setQuickAccessButtonHeight()` (*ribbon.titlewidget.RibbonTitleWidget method*), 19

`setRibbonHeight()` (*ribbon.ribbonbar.RibbonBar method*), 15

`setRibbonStyle()` (*ribbon.ribbonbar.RibbonBar method*), 15

`setRibbonVisible()` (*ribbon.ribbonbar.RibbonBar method*), 15

`setRightToolBarHeight()` (*ribbon.ribbonbar.RibbonBar method*), 15

`setRightToolBarHeight()` (*ribbon.titlewidget.RibbonTitleWidget method*), 19

`setSelectedButton()` (*ribbon.gallery.RibbonGallery method*), 37

`setTabBarHeight()` (*ribbon.ribbonbar.RibbonBar method*), 15

`setTabBarHeight()` (*ribbon.titlewidget.RibbonTitleWidget method*), 19

`setTitle()` (*ribbon.ribbonbar.RibbonBar method*), 15

`setTitle()` (*ribbon.titlewidget.RibbonTitleWidget method*), 19

`setTitleText()` (*ribbon.panel.RibbonPanel method*), 34

`setTopBottomMargins()` (*ribbon.separator.RibbonSeparator method*), 39

`showContextCategory()` (*ribbon.category.RibbonContextCategory method*), 22

`showContextCategory()` (*ribbon.ribbonbar.RibbonBar method*), 15

`showPopup()` (*ribbon.gallery.RibbonGallery method*), 37

`showRibbon()` (*ribbon.ribbonbar.RibbonBar method*), 15

`sizeHint()` (*ribbon.separator.RibbonSeparator method*), 39

`tabBar()` (*ribbon.titlewidget.RibbonTitleWidget method*), 19

`tabBarHeight()` (*ribbon.ribbonbar.RibbonBar method*), 15

`tabBarHeight()` (*ribbon.titlewidget.RibbonTitleWidget method*), 19

`tabRect()` (*ribbon.ribbonbar.RibbonBar method*), 16

`takePanel()` (*ribbon.category.RibbonCategory method*), 21

`takeWidget()` (*ribbon.categorylayoutwidget.RibbonCategoryLayoutWidget method*), 24

`title()` (*ribbon.category.RibbonCategory method*), 21

`title()` (*ribbon.ribbonbar.RibbonBar method*), 16

`title()` (*ribbon.titlewidget.RibbonTitleWidget method*), 19

`titleText()` (*ribbon.panel.RibbonPanel method*), 35

W

`widget()` (*ribbon.panel.RibbonPanel method*), 35

T

`tabBar()` (*ribbon.ribbonbar.RibbonBar method*), 15