pyqtribbon

Release 0.6.1

WANG Hailin

Mar 26, 2023

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 User Manual **7**
- 3.1 The RibbonScreenShotWindow Class . . . 7
- 3.2 Instantiate a Ribbon Bar . . . 7
- 3.3 Customize Ribbon Bar . . . 8
- 3.4 Customize Categories . . . 11
- 3.5 Customize Panels . . . 13
- 3.6 A Complete Example . . . 16

4 API Reference **19**

5 Indices and tables **21**

Ribbon Bar for PyQt or PySide applications.

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

CHAPTER

ONE

GETTING STARTED

1.1 Installation

PyQtRibbon is distribued 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
```

CHAPTER

TWO

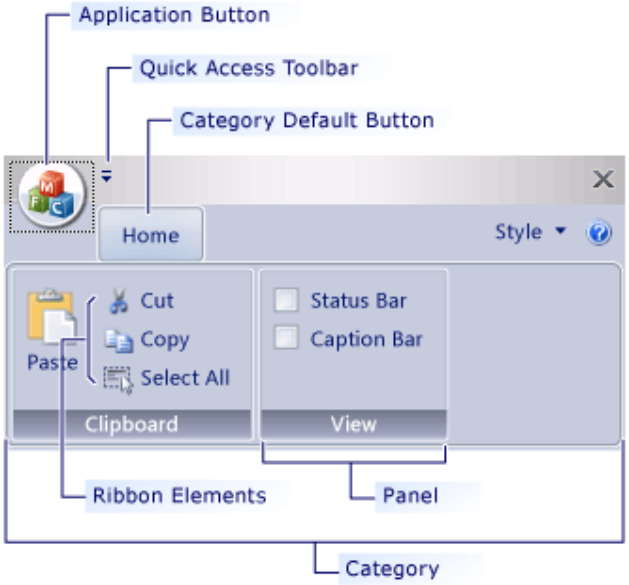
THE RIBBON BAR

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:

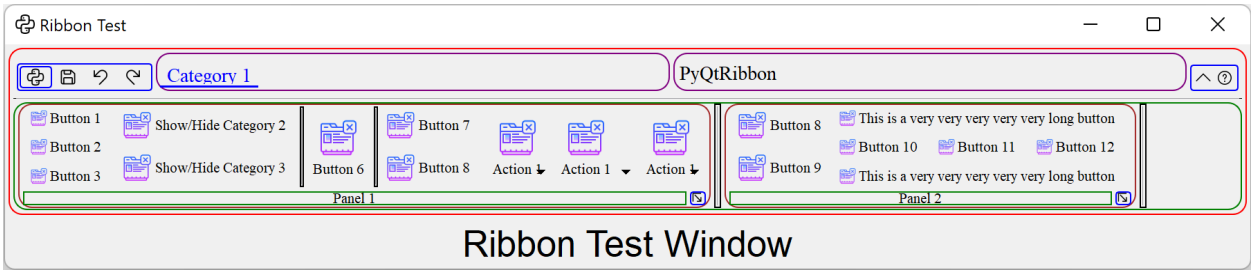
Note: A ribbon is a user interface (UI) element that organizes commands into logical groups. These groups appear on separate tabs in a strip across the top of the window. The ribbon replaces the menu bar and toolbars. A ribbon can significantly improve application usability. For more information, see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



CHAPTER

THREE

USER MANUAL

3.1 The RibbonScreenShotWindow Class

The `RibbonScreenShotWindow` class is just for taking a screenshot of the window, the window will be closed 0.1s after it is shown. It is just used for documenting the window.

class pyqtribbon.screenshotwindow.**RibbonScreenShotWindow**(*fileName: str = 'shot.jpg'*, **args*, ***kwargs*)

This class is just for taking a screenshot of the window, the window will be closed 0.1s after it is shown.

Initialize the class.

Parameters
fileName – The file name for the screenshot.

setScreenShotFileName(*fileName: str*)

Set the file name for the screenshot.

Parameters
fileName – The file name for the screenshot.

takeScreenShot()

Take a screenshot of the window.

3.2 Instantiate a Ribbon Bar

`RibbonBar` is inherited from `QMenuBar`, you can use the *setMenuBar* method of `QMainWindow` to set the ribbon bar as the main menu bar.

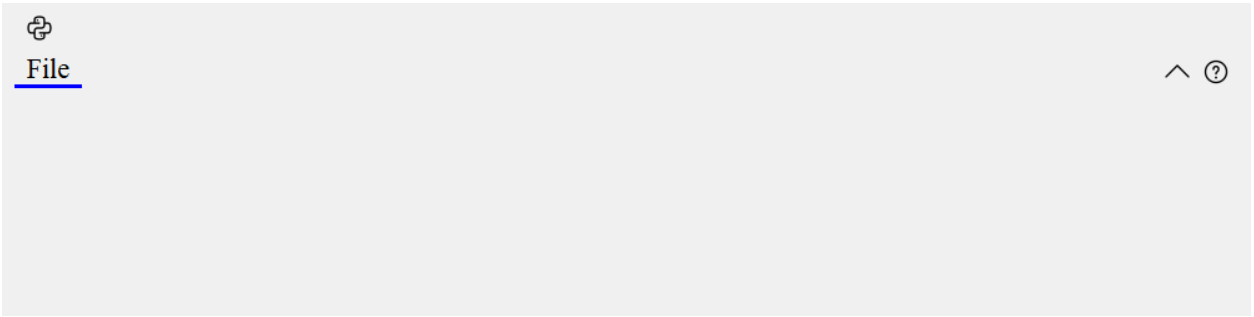
```
from pyqtribbon import RibbonBar

window = QtWidgets.QMainWindow()
ribbon = RibbonBar()
window.setMenuBar(ribbon)
```

3.2.1 Example

For example, using the following code,

```
import sys

from qtpy import QtWidgets, QtGui

from pyqtribbon import RibbonBar
from pyqtribbon.screenshotwindow import RibbonScreenShotWindow

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

    # Ribbon bar
    ribbonbar = RibbonBar()
    window.setMenuBar(ribbonbar)

    # Show the window
    window.resize(1000, 250)
    window.show()

    sys.exit(app.exec_())
```

You can get a window like this:



3.3 Customize Ribbon Bar

3.3.1 General Setups

`RibbonBar.setRibbonStyle`(style)	Set the style of the ribbon.
`RibbonBar.ribbonHeight`()	Get the total height of the ribbon.
`RibbonBar.setRibbonHeight`(height)	Set the total height of the ribbon.
`RibbonBar.showRibbon`()	Show the ribbon.
`RibbonBar.hideRibbon`()	Hide the ribbon.
`RibbonBar.ribbonVisible`()	Get the visibility of the ribbon.
`RibbonBar.setRibbonVisible`(visible)	Set the visibility of the ribbon.

3.3.2 Setup Application Button

RibbonBar.applicationOptionButton()	Return the application button.
RibbonBar.setApplicationIcon(icon)	Set the application icon.
RibbonBar.addFileMenu()	Add a file menu to the ribbon.

3.3.3 Setup Title

RibbonBar.title()	Return the title of the ribbon.
RibbonBar.setTitle(title)	Set the title of the ribbon.
RibbonBar.addTitleWidget(widget)	Add a widget to the title widget.
RibbonBar.insertTitleWidget(index, widget)	Insert a widget to the title widget.
RibbonBar.removeTitleWidget(widget)	Remove a widget from the title widget.

3.3.4 Setup Category Tab Bar

RibbonBar.tabBar()	Return the tab bar of the ribbon.

3.3.5 Setup Quick Access Bar

RibbonBar.quickAccessToolBar()	Return the quick access toolbar of the ribbon.
RibbonBar.addQuickAccessButton(button)	Add a button to the quick access bar.
RibbonBar.setQuickAccessButtonHeight([height])	Set the height of the quick access buttons.

3.3.6 Setup Right Tool Bar

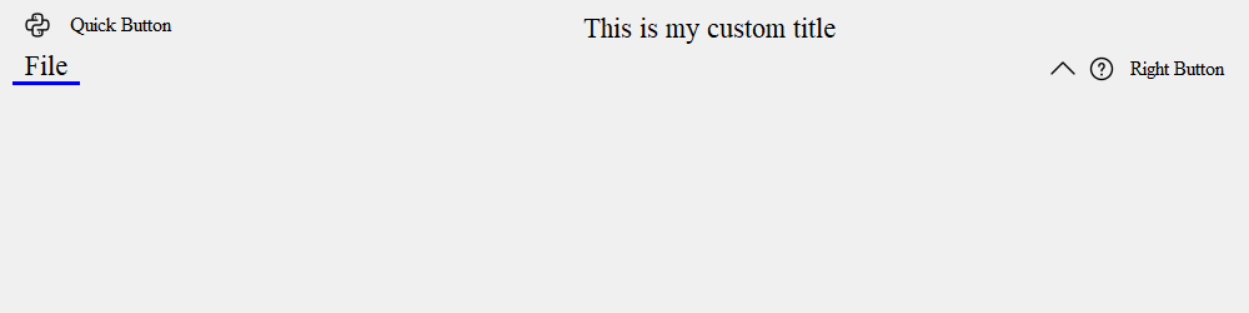
RibbonBar.rightToolBar()	Return the right toolbar of the ribbon.
RibbonBar.addRightToolButton(button)	Add a widget to the right button bar.
RibbonBar.setRightToolBarHeight([height])	Set the height of the right buttons.
RibbonBar.setHelpButtonIcon(icon)	Set the icon of the help button.
RibbonBar.removeHelpButton()	Remove the help button from the ribbon.
RibbonBar.helpButtonClicked(bool)	Signal, the help button was clicked.
RibbonBar.collapseRibbonButton()	Return the collapse ribbon button.
RibbonBar.setCollapseButtonIcon(icon)	Set the icon of the min button.
RibbonBar.removeCollapseButton()	Remove the min button from the ribbon.

3.3.7 Example

For example, using the following code,

```
import sys

from qtpy import QtGui
from qtpy.QtWidgets import QApplication, QToolButton

from pyqtribbon import RibbonBar
from pyqtribbon.screenshotwindow import RibbonScreenShotWindow

if __name__ == '__main__':
    app = QApplication(sys.argv)
    app.setFont(QtGui.QFont("Times New Roman", 8))
    window = RibbonScreenShotWindow('ribbonbar-customize.png')

    # Ribbon bar
    ribbonbar = RibbonBar()
    window.setMenuBar(ribbonbar)

    # Title of the ribbon
    ribbonbar.setTitle('This is my custom title')

    # Quick Access Bar
    qbutton = QToolButton()
    qbutton.setText('Quick Button')
    ribbonbar.addQuickAccessButton(qbutton)

    # Right toolbar
    rbutton = QToolButton()
    rbutton.setText('Right Button')
    ribbonbar.addRightToolButton(rbutton)

    # Show the window
    window.resize(1000, 250)
    window.show()

    sys.exit(app.exec_())
```

You can get a window like this:



3.3.8 Manage Categories

RibbonBar.categories()	Return a list of categories of the ribbon.
RibbonBar.addCategory(title[, style, color])	Add a new category to the ribbon.
RibbonBar.addCategoriesBy(data)	Add categories from a dict.
RibbonBar.addNormalCategory(title)	Add a new category to the ribbon.
RibbonBar.addContextCategory(title[, color])	Add a new context category to the ribbon.
RibbonBar.addContextCategories(name, titles)	Add a group of context categories with the same tab color to the ribbon.
RibbonBar.showContextCategory(category)	Show the given category or categories, if it is not a context category, nothing happens.
RibbonBar.hideContextCategory(category)	Hide the given category or categories, if it is not a context category, nothing happens.
RibbonBar.removeCategory(category)	Remove a category from the ribbon.
RibbonBar.setCurrentCategory(category)	Set the current category.
RibbonBar.currentCategory()	Return the current category.
RibbonBar.showCategoryByIndex(index)	Show category by tab index

3.4 Customize Categories

3.4.1 Setup Styles

RibbonCategory.categoryStyle()	Return the button style of the category.
RibbonCategory.setCategoryStyle(style)	Set the button style of the category.

3.4.2 Manage Panels

RibbonCategory.addPanel(title[, ...])	Add a new panel to the category.
RibbonCategory.addPanelsBy(data)	Add panels from a dictionary.
RibbonCategory.removePanel(title)	Remove a panel from the category.
RibbonCategory.takePanel(title)	Remove and return a panel from the category.
RibbonCategory.panel(title)	Return a panel from the category.
RibbonCategory.panels()	Return all panels in the category.

3.4.3 Example

For example, using the following code,

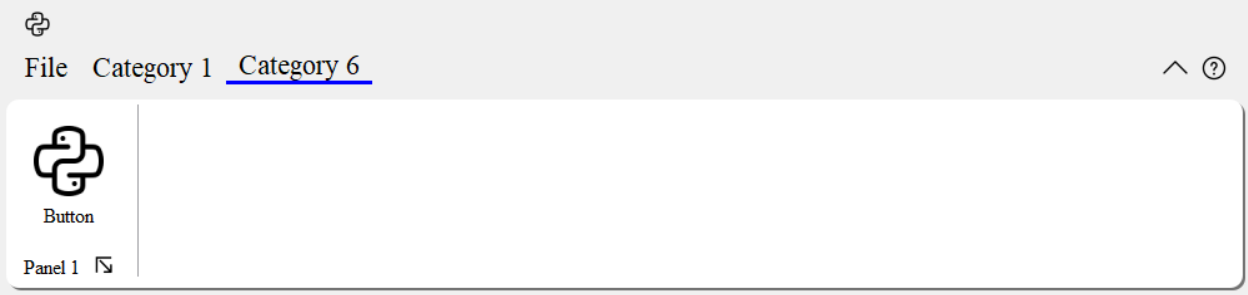
```
import sys

from qtpy import QtGui
from qtpy.QtWidgets import QApplication
from qtpy.QtGui import QIcon

from pyqtribbon import RibbonBar, RibbonCategoryStyle
```

(continues on next page)

(continued from previous page)

```
from pyqtribbon.screenshotwindow import RibbonScreenShotWindow

if __name__ == '__main__':
    app = QApplication(sys.argv)
    app.setFont(QtGui.QFont("Times New Roman", 8))
    window = RibbonScreenShotWindow('category.png')

    # Ribbon bar
    ribbonbar = RibbonBar()
    window.setMenuBar(ribbonbar)

    # Categories
    category1 = ribbonbar.addCategory('Category 1')
    panel1 = category1.addPanel('Panel 1')
    panel1.addLargeButton('Large Button 1', QIcon('python.png'))

    category2 = ribbonbar.addContextCategory('Category 2')
    panel12 = category2.addPanel('Panel 2')
    panel12.addLargeButton('Large Button 2', QIcon('python.png'))

    categories = ribbonbar.addCategoriesBy({
        'Category 6': {
            "style": RibbonCategoryStyle.Normal,
            "panels": {
                "Panel 1": {
                    "showPanelOptionButton": True,
                    "widgets": {
                        "Button 1": {
                            "type": "Button",
                            "arguments": {
                                "icon": QIcon("python.png"),
                                "text": "Button",
                                "tooltip": "This is a tooltip",
                            }
                        },
                    }
                },
            }
        }
    })
    ribbonbar.setCurrentCategory(categories['Category 6'])

    # Show the window
    window.resize(1000, 250)
    window.show()

    sys.exit(app.exec_())
```

You can get a window like this:



3.5 Customize Panels

3.5.1 Setup Title Label

`RibbonPanel.title`()	Get the title of the panel.
`RibbonPanel.setTitle`(title)	Set the title of the panel.

3.5.2 Setup Panel Option Button

`RibbonPanel.panelOptionButton`()	Return the panel option button.
`RibbonPanel.setPanelOptionToolTip`(text)	Set the tooltip of the panel option button.
`RibbonPanel.panelOptionClicked`(bool)	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

3.5.3 Add Widgets to Panels

`RibbonPanel.addWidget`(widget, *[, rowSpan, ...])	Add a widget to the panel.
`RibbonPanel.addWidgetsBy`(data)	Add widgets to the panel.
`RibbonPanel.removeWidget`(widget)	Remove a widget from the panel.
`RibbonPanel.widget`(index)	Get the widget at the given index.
`RibbonPanel.widgets`()	Get all the widgets in the panel.
`RibbonPanel.addSmallWidget`(widget, *[, ...])	
`RibbonPanel.addMediumWidget`(widget, *[, ...])	
`RibbonPanel.addLargeWidget`(widget, *[, ...])	
`RibbonPanel.addButton`([text, icon, ...])	Add a button to the panel.
`RibbonPanel.addSmallButton`([text, icon, ...])	
`RibbonPanel.addMediumButton`([text, icon, ...])	
`RibbonPanel.addLargeButton`([text, icon, ...])	
`RibbonPanel.addToggleButton`([text, icon, ...])	

continues on next page

Table 1 – continued from previous page

RibbonPanel.addSmallToggleButton([text, ...])	
RibbonPanel.addMediumToggleButton([text, ...])	
RibbonPanel.addLargeToggleButton([text, ...])	
RibbonPanel.addComboBox(*args[, cls, ...])	
RibbonPanel.addFontComboBox(*args[, cls, ...])	
RibbonPanel.addLineEdit(*args[, cls, ...])	
RibbonPanel.addTextEdit(*args[, cls, ...])	
RibbonPanel.addPlainTextEdit(*args[, cls, ...])	
RibbonPanel.addLabel(*args[, cls, initializer])	
RibbonPanel.addProgressBar(*args[, cls, ...])	
RibbonPanel.addSlider(*args[, cls, initializer])	
RibbonPanel.addSpinBox(*args[, cls, initializer])	
RibbonPanel.addDoubleSpinBox(*args[, cls, ...])	
RibbonPanel.addDateEdit(*args[, cls, ...])	
RibbonPanel.addTimeEdit(*args[, cls, ...])	
RibbonPanel.addDateTimeEdit(*args[, cls, ...])	
RibbonPanel.addTableWidget(*args[, cls, ...])	
RibbonPanel.addTreeWidget(*args[, cls, ...])	
RibbonPanel.addListWidget(*args[, cls, ...])	
RibbonPanel.addCalendarWidget(*args[, cls, ...])	
RibbonPanel.addSeparator([orientation, width])	Add a separator to the panel.
RibbonPanel.addHorizontalSeparator(*[, ...])	
RibbonPanel.addVerticalSeparator(*[, ...])	
RibbonPanel.addGallery([minimumWidth, ...])	Add a gallery to the panel.

3.5.4 Example

For example, using the following code,

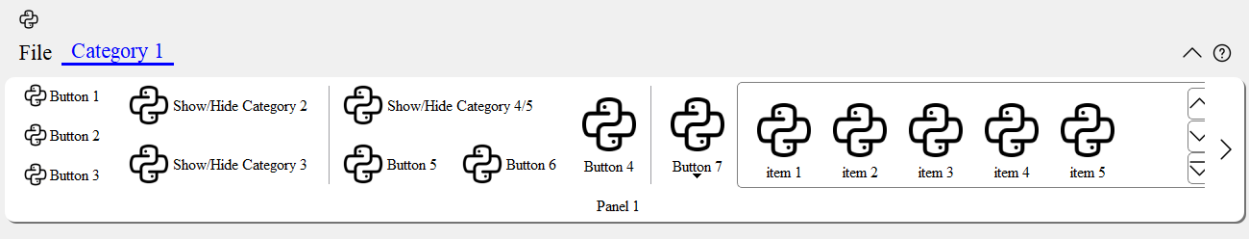
```
import sys

from qtpy import QtGui
from qtpy.QtWidgets import QApplication, QToolButton, QMenu, QLabel, QLineEdit
from qtpy.QtGui import QIcon
from qtpy.QtCore import Qt

from pyqtribbon import RibbonBar
from pyqtribbon.screenshotwindow import RibbonScreenShotWindow

if __name__ == '__main__':
    app = QApplication(sys.argv)
    app.setFont(QtGui.QFont("Times New Roman", 8))
    window = RibbonScreenShotWindow('panel.png')

    # Ribbon bar
    ribbonbar = RibbonBar()
    window.setMenuBar(ribbonbar)

    category1 = ribbonbar.addCategory("Category 1")
    panel = category1.addPanel("Panel 1", showPanelOptionButton=False)
    panel.addSmallButton("Button 1", icon=QIcon("python.png"))
    panel.addSmallButton("Button 2", icon=QIcon("python.png"))
    panel.addSmallButton("Button 3", icon=QIcon("python.png"))
    panel.addMediumToggleButton("Show/Hide Category 2", icon=QIcon("python.png"))
    panel.addVerticalSeparator()
    panel.addMediumToggleButton("Show/Hide Category 3", icon=QIcon("python.png"))
    panel.addMediumToggleButton("Show/Hide Category 4/5", icon=QIcon("python.png"),␣
↪colSpan=2, alignment=Qt.AlignLeft)
    panel.addLargeButton("Button 4", icon=QIcon("python.png"))
    panel.addVerticalSeparator()
    panel.addMediumButton("Button 5", icon=QIcon("python.png"))
    panel.addMediumButton("Button 6", icon=QIcon("python.png"))

    button = panel.addLargeButton("Button 7", icon=QIcon("python.png"))
    menu = QMenu()
    menu.addAction(QIcon("python.png"), "Action 1")
    menu.addAction(QIcon("python.png"), "Action 2")
    menu.addAction(QIcon("python.png"), "Action 3")
    button.setMenu(menu)
    button.setPopupMode(QToolButton.InstantPopup)
    panel.addWidget(button, rowSpan=6)

    gallery = panel.addGallery(minimumWidth=500, popupHideOnClick=True)
    for i in range(100):
        gallery.addToggleButton(f'item {i+1}', QIcon("python.png"))
    popupMenu = gallery.popupMenu()
    submenu = popupMenu.addMenu(QIcon("python.png"), 'Submenu')
    submenu.addAction(QIcon("python.png"), "Action 4")
```

(continues on next page)

(continued from previous page)

```
    popupMenu.addAction(QIcon("python.png"), "Action 1")
    popupMenu.addAction(QIcon("python.png"), "Action 2")
    popupMenu.addSeparator()
    popupMenu.addWidget(QLabel("This is a custom widget"))
    formLayout = popupMenu.addFormLayoutWidget()
    formLayout.addRow(QLabel("Row 1"), QLineEdit())

    # Show the window
    window.resize(1300, 250)
    window.show()

    sys.exit(app.exec_())
```

You can get a window like this:



3.6 A Complete Example

The following code snippet is a complete example.

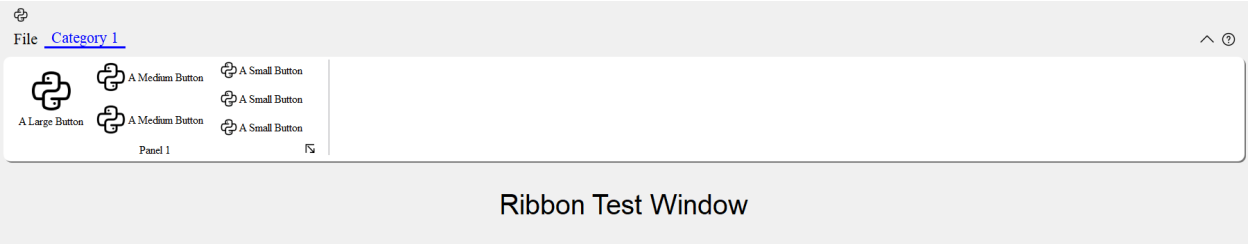
```
import sys

from PyQt5.QtWidgets import QApplication, QLabel, QWidget, QVBoxLayout
from PyQt5.QtGui import QIcon, QFont
from PyQt5.QtCore import Qt

from pyqtribbon import RibbonBar
from pyqtribbon.screenshotwindow import RibbonScreenShotWindow
from pyqtribbon.utils import DataFile

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

    # Central widget
    window = RibbonScreenShotWindow('tutorial-ribbonbar.png')
    window.setWindowIcon(QIcon(DataFile("icons/python.png")))
    centralWidget = QWidget()
    window.setCentralWidget(centralWidget)
    layout = QVBoxLayout(centralWidget)

    # Ribbon bar
    ribbonbar = RibbonBar()
```

(continues on next page)

(continued from previous page)

```
    window.setMenuBar(ribbonbar)
    category = ribbonbar.addCategory("Category 1")
    panel = category.addPanel("Panel 1")
    panel.addLargeButton("A Large Button", QIcon(DataFile("icons/python.png")))
    panel.addMediumButton("A Medium Button", QIcon(DataFile("icons/python.png")))
    panel.addMediumButton("A Medium Button", QIcon(DataFile("icons/python.png")))
    panel.addSmallButton("A Small Button", QIcon(DataFile("icons/python.png")))
    panel.addSmallButton("A Small Button", QIcon(DataFile("icons/python.png")))
    panel.addSmallButton("A Small Button", QIcon(DataFile("icons/python.png")))

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

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

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

You can get a window like this:



CHAPTER

FOUR

API REFERENCE

This page contains auto-generated API reference documentation[1].

[1] Created with sphinx-autoapi

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search