Chapter 10 – Toolbars

Toolbardemo application

Step 1 Open Qt Designer

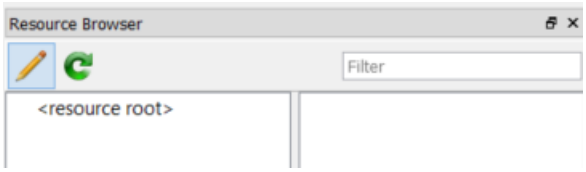
- Open the QtDesigner by selecting the Qt Designer program icon

Step 2 Selecting a template

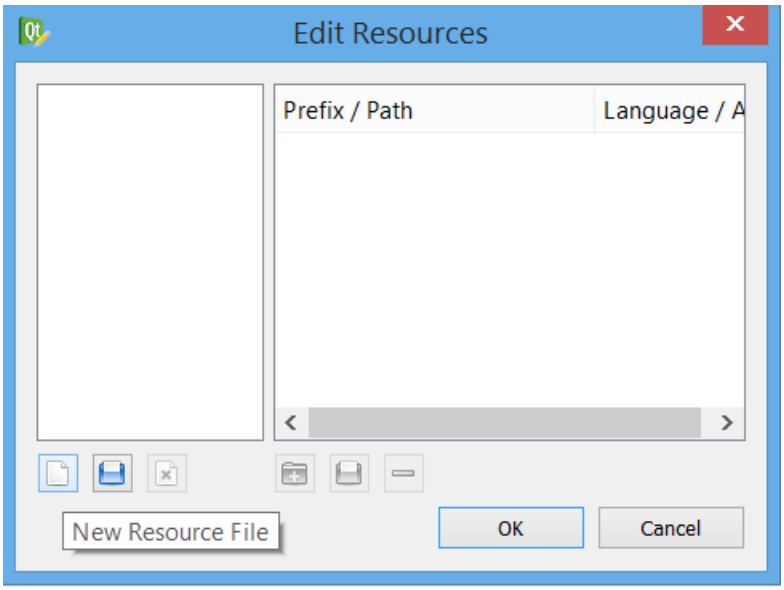
- Click on the “Main Window”
- Click the Create button
- A new form with the caption “untitled” is created with a menu

Step 3 Create a resource file

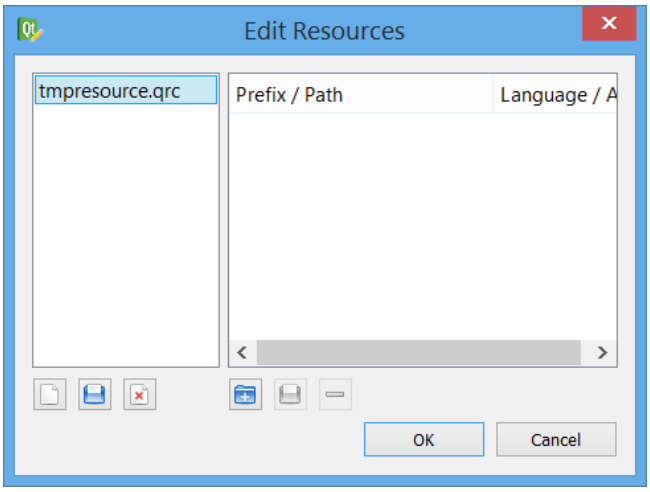
- We will create a resource file that contains the icons for the toolbar.
- Copy the .ico files provided to your computer
- Open the Resource Browser by clicking on the Resource Browser tab at the bottom right of the screen
- Click the edit resource icon



- Create a new resource file by clicking on the New Resource file button

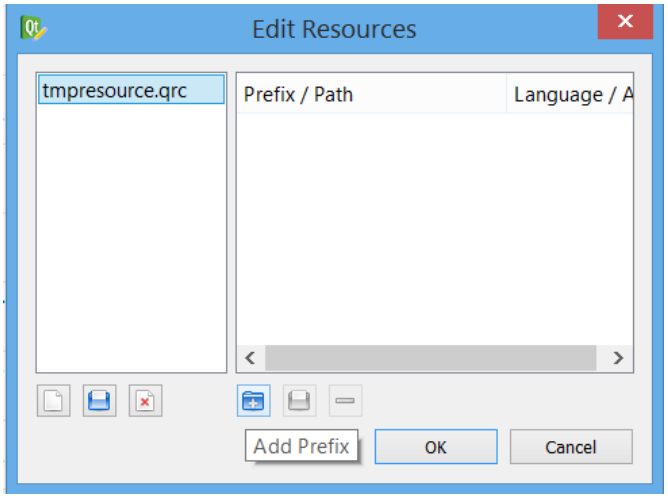


- Enter a filename, tmpresource and click Save

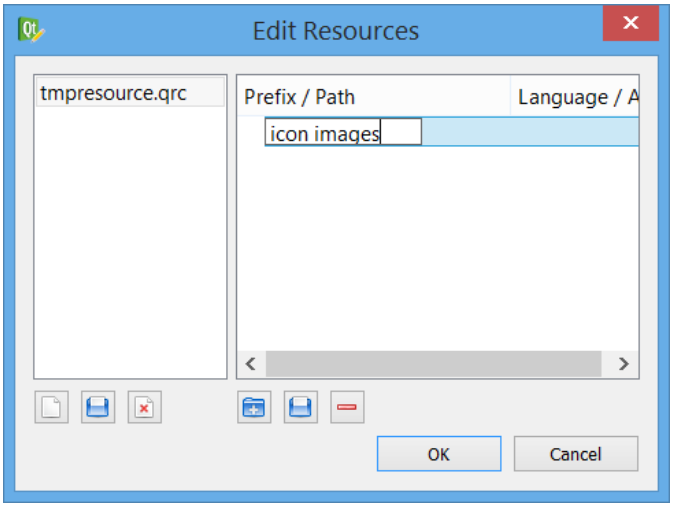


Step 4 Add resources to the resource file

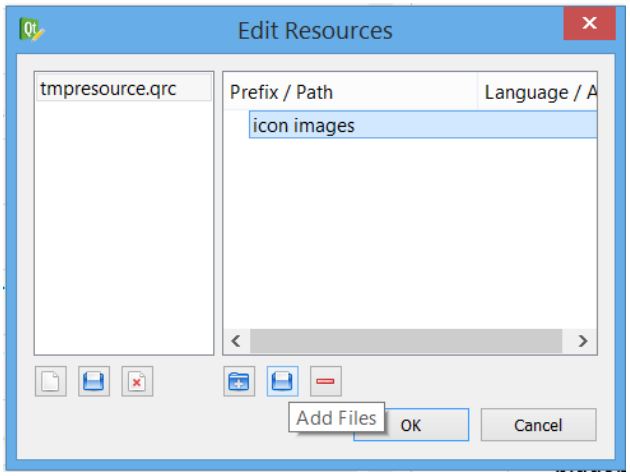
- To add a resource, we first need to create a prefix, a prefix is category name for a resource.
- Click the Add prefix button



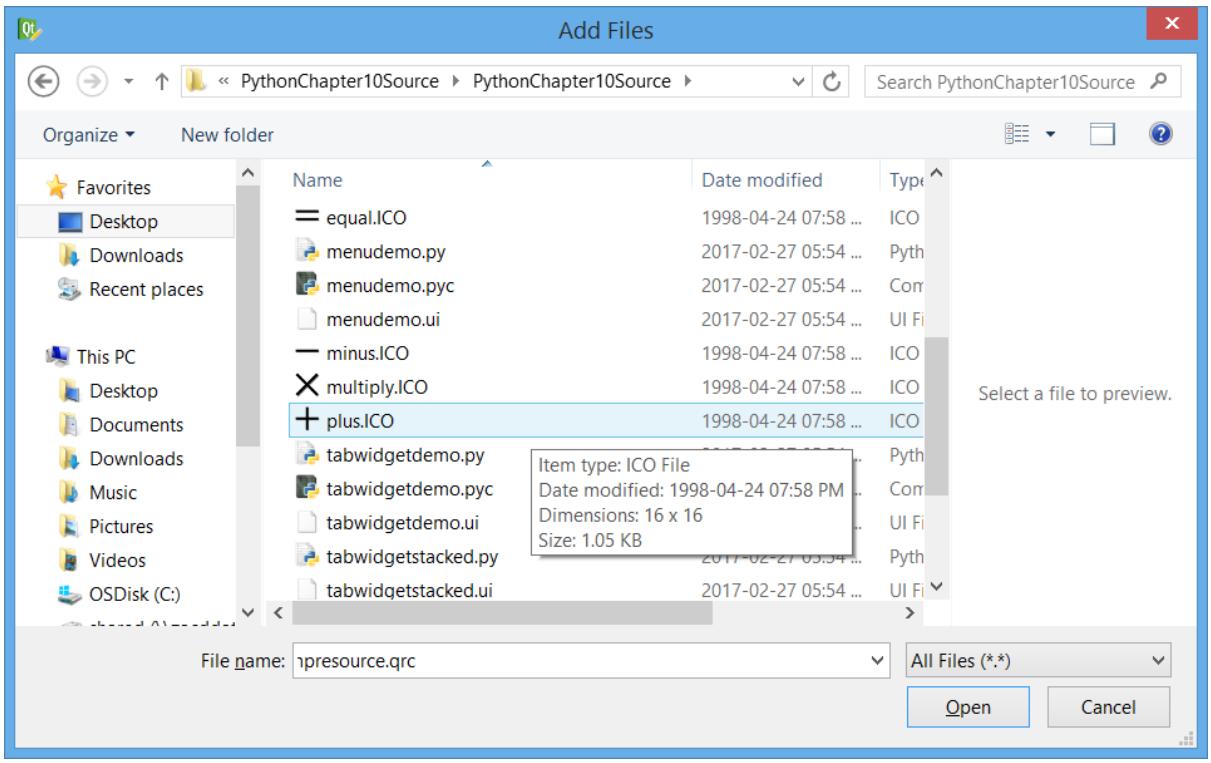
- Change the name to **icon images**



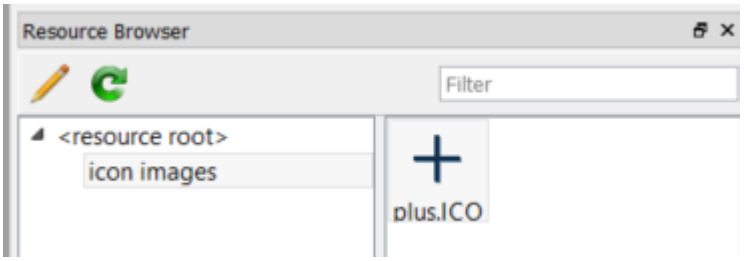
- Click on the Add files icon



- Browse to the location where you save the .ico files and select the file `plus.ico` and click open



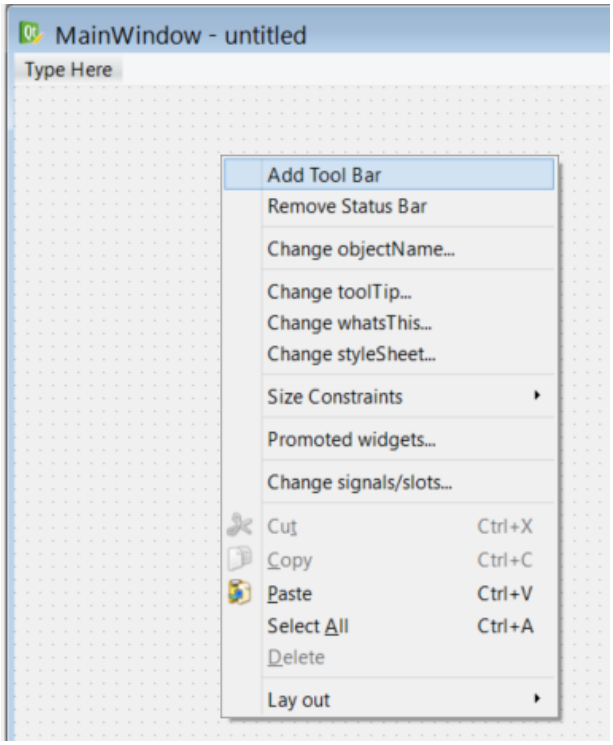
- Click OK
- The icon should be added to your resource file under the icon images prefix



- Repeat the process for the minus, multiply, divide and equal icons

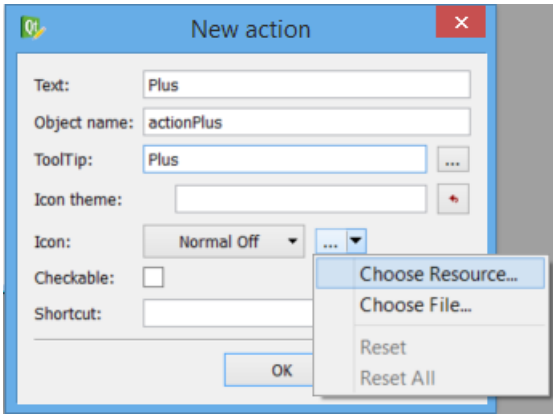
Step 4 Adding a toolbar

- Right click on the Main Window
- Select Add Tool Bar from the menu

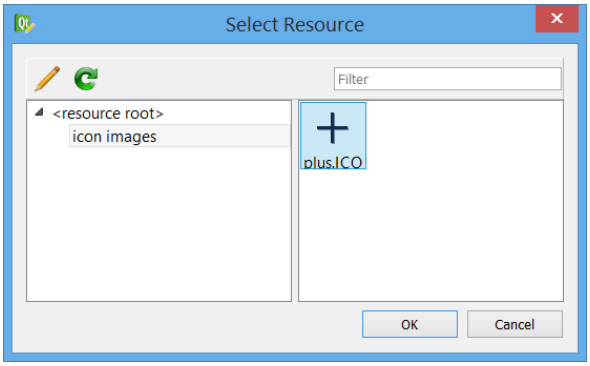


- A blank toolbar will be added below the menu bar
- Toolbar buttons are added with actions, we will therefor create an action in the Action editor for each toolbar button and drag each action from the Action Editor and drop it on the toolbar
- Open the Action Editor by clicking on the tab at the bottom right of the screen.
- We will create a toolbar with icons for arithmetic operators
- Click on the new button on the action editor

- In the Text box specify the name of the action, **Plus**
- The Objectname will automatically be updated to action with the text as a prefix, actionPlus
- In the Tooltip box enter Plus or a description of the action
- On the icon dropdown list select the option **Choose Resource**



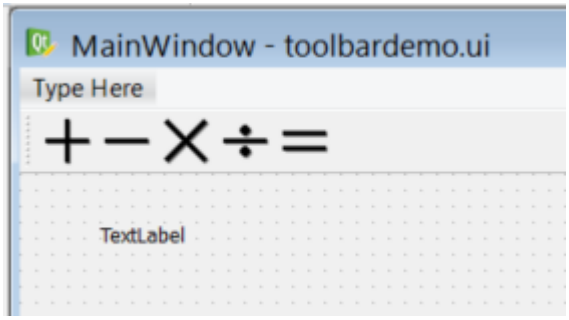
- Click on the plus.ico and then OK to assign the icon to the action



- Drag the actionPlus from the Action Editor onto the toolbar
- A new icon is created on the toolbar which is linked to the actionPlus action
- Repeat the process for Minus, Multiply, Divide and Equal

Step 5 Add a label

- Add a label widget on the form and change the name to `label`



Step 6 Save the form

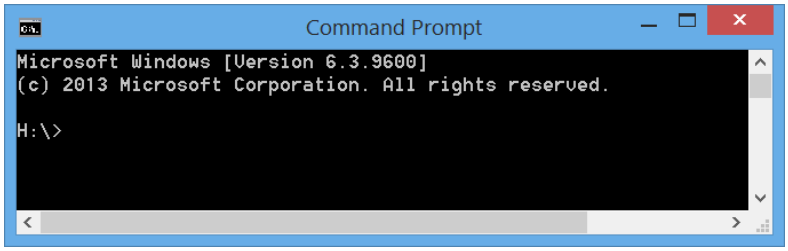
- Save the form as `toolbardemo.ui` **(note the case!! Python is case sensitive)**

Step 7 Convert the .ui file to a .py file

- Convert the `toolbardemo.ui` file to `toolbardemo.py` using `pyuic4`. **(note the case!! Python is case sensitive, so even on file names the case must be the same throughout)**

Step 8 Convert the .qrc file to a .py file

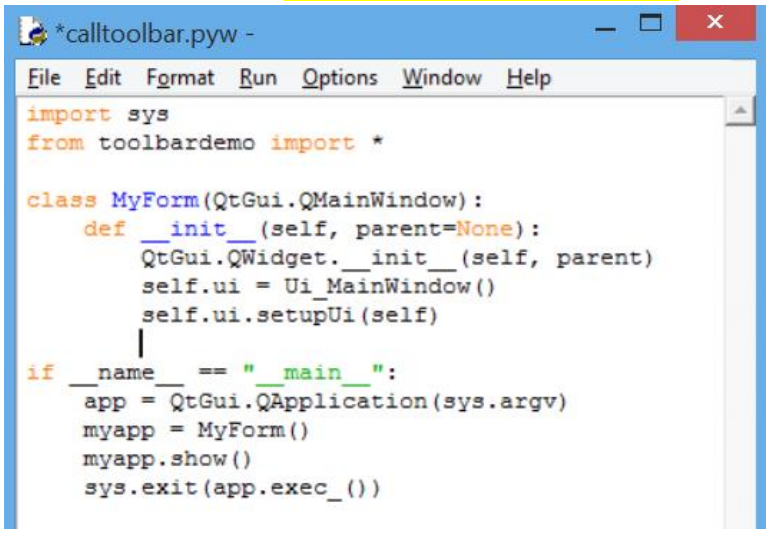
- The resource file needs to be converted to a Python .py file as well.
- We will use another command prompt utility for converting the .qrc file to a Python script
- Make sure that you copy the resource .qrc file and the icon files to the `C:\Python34\Lib\site-packages\PyQt4` folder
- Click the windows Start button and look command prompt or type cmd to open the command prompt window



- Type `cd C:\Python34\Lib\site-packages\PyQt4` and press Enter
- Type `pyrcc4 -py3 tmpresource.qrc -o tmpresource_rc.py` and press Enter **(note the case!! Python is case sensitive, so even on file names the case must be the same throughout)**
- The screen will pause for a while and return a blank line if the conversion was successful
- Copy the `tmpresource.qrc` and `tmpresource_rc.py` files from the `C:\Python34\Lib\site-packages\PyQt4` folder to a new folder for this application under your documents

Step 9 Create a source file (.pyw) that imports the .py file

- Create a source file that will import the .py file created in step above and from which we will invoke the user interface
- Use the following code (note the indentation and case!!)



- Save the file as `calltoolbar.pyw`
- Run and test the application up to this point

Step 9 Add the code

- Write functions for each of the toolbar icons
- Connect the triggered() signal of each of the menu items to the methods you created
- Add the following code: (Note the indentation and case in the screenprint)

```
        self.connect(self.ui.actionPlus, QtCore.SIGNAL('triggered()'),
self.plusmessage)
        self.connect(self.ui.actionMinus, QtCore.SIGNAL('triggered()'),
self.minusmessage)
        self.connect(self.ui.actionMultiply,
QtCore.SIGNAL('triggered()'), self.multiplymessage)
        self.connect(self.ui.actionDivide, QtCore.SIGNAL('triggered()'),
self.dividemessage)
        self.connect(self.ui.actionEqual, QtCore.SIGNAL('triggered()'),
self.equalmessage)

    def plusmessage(self):
        self.ui.label.setText("You have selected Plus ")

    def minusmessage(self):
        self.ui.label.setText("You have selected Minus ")

    def multiplymessage(self):
        self.ui.label.setText("You have selected Multiply ")

    def dividemessage(self):
        self.ui.label.setText("You have selected Divide ")

    def equalmessage(self):
        self.ui.label.setText("You have selected Equal ")
```

calltoolbar.pyw -

File Edit Format Run Options Window Help

```
import sys
from toolbardemo import *

class MyForm(QtGui.QMainWindow):
    def __init__(self, parent=None):
        QtGui.QWidget.__init__(self, parent)
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.connect(self.ui.actionPlus, QtCore.SIGNAL('triggered()'), self.plusmessage)
        self.connect(self.ui.actionMinus, QtCore.SIGNAL('triggered()'), self.minusmessage)
        self.connect(self.ui.actionMultiply, QtCore.SIGNAL('triggered()'), self.multiplymessage)
        self.connect(self.ui.actionDivide, QtCore.SIGNAL('triggered()'), self.dividemessage)
        self.connect(self.ui.actionEqual, QtCore.SIGNAL('triggered()'), self.equalmessage)

    def plusmessage(self):
        self.ui.label.setText("You have selected Plus ")

    def minusmessage(self):
        self.ui.label.setText("You have selected Minus ")

    def multiplymessage(self):
        self.ui.label.setText("You have selected Multiply ")

    def dividemessage(self):
        self.ui.label.setText("You have selected Divide ")

    def equalmessage(self):
        self.ui.label.setText("You have selected Equal ")

if __name__ == "__main__":
    app = QtGui.QApplication(sys.argv)
    myapp = MyForm()
    myapp.show()
    sys.exit(app.exec_())
```

- Save the file as `calltoolbar.pyw`
- Run and test your application. Click each toolbar icon and see that it changes the label text