Quick Start

from November 3, 2010

Here it comes the first small example written in Q7Basic by you. All you need to do is to follow the following steps and read the explanations. It will give you some feeling about writing in Q7Basic and Qt – a whole new world to explore.

You will need some time to do the task. Plan to have time for about an hour before you have completed this quick start.

Let's go

Start Q7Basic by double clicking on the symbol.

Create a new project

Select in the menubar *File -> New Project...* Enter a short name used for your project directory and your project overall. Don't use / . or other special characters. What about the name *Quick Start*?

After pressing OK, a new skeleton project will be created for you (normally on the Desktop of the current user).

The file G**`lobal.QObject`** (normally just called **Global**) is a special file created automatically for you with the special procedure **`Event Init()`**. The Global.QObject file is actually the place where you may declare variables, constants, functions and so on, which may be used as global elements of your program without the need of an object creation before. It is similar to a module in VB. The Event Init() is called on startup of your application. You may freely add code to it. But we won't do much with it in this small tutorial.

First start of your application

Anyway, let's just run this small example and take a look what happens when you hit the run button (in the toolbar of the Q7Basic IDE or in the menubar *Compiler -> Run)*.

You probably noticed the command MsgBox **`("Init")`** in the Event Init(), which makes the window appearing with the texts **`Init`** after you started your example.

Stopping your running example

When you hit the Stop button in the toolbar or in the menubar *Compiler -> Stop,* your application will immediately stop execution.

First change of your application
Let us change the “Init” to something more meaningful like "Hello World!”. To do so write MsgBox("Hello World!”) between Event Init and End Event. Run the example again, if you don't know how, see the paragraph above. Now, Hello World!, appears.

Hey, you just change your first application in Q7Basic. Congratulations!

Windows and buttons
Time to get a more sophisticated way to get a window on screen. Now, I want you to start **Qt Designer** create a new main window, with a button on it. Change the button's title when clicking on it and show a message when it is pressed.

Qt Designer is our friend. To start it, make the code file MainWindow.QMainWindow the current file (its code must be visible) and select in the menubar *Project -> Qt Designer For Current File.*

Qt Designer gets loaded with the default GUI file MainWindow.ui.

Qt Designer

To get a quick overview about Qt Designer, read the documentation provided for it, called "Qt Designer” selectable in the menubar in the help menu of the Q7Basic IDE.

You are in Qt Designer now:

New Window
After you opened Qt Desigern, the main window will appear as empty window on screen ready to be exteneded by dropping controls on it.

New QPushButton
Place a `QPushButton` on the window by drag & drop it to the new created window. (Press the left button on the Push Button and draw your mouse onto the window and stop pressing the mouse button.)

Basic usage of Qt Designer
In Qt Designer you draw your windows and controls, but you do not code there (do not forget to give all objects meaningful object names).

Next Step
Save the file opened in Qt Designer. In the menubar of Qt Designer : *File -> Save*

Now, you have created a window and a button on that window, you need to tell the Q7Basic IDE what have to be done with them. But before that, it is time to run your example again. Change to IDE of Q7Basic by clicking on the icon on the bottom of your screen and select run again.

What will happen? Your message will appear again, but after clicking on OK, your newly created window with the button will appear. Great! You managed to get a custom Qt based Window to be shown on your screen.

Writing code for the window

We need a place to write code for the window. Code is always written in the IDE of Q7Basic. We do not need to create a new source code file in the IDE of Q7Basic, because MainWindow.QMainWindow can be used. (If you would need a new code file. You could create it by selecting in the menubar Project -> New File With Super Class -> QObject... enter *code* as name.)

In MainWindow.QMainWindow add a new line:

```
Outlet pushButton As QPushButton ' if your button is not named pushButton,
change it!
```

Where you add this line is regardless.
It declares a variable used for a button in a ui file, which can be accessed and used in source code. Within your project, it is used for the button in the newly created window in the ui file.

After that create the following lines as well.

```
Signal on_pushButton_triggered(Checked As Boolean) ' if your button is not named
pushButton, change it!
  pushButton.Text = "Just changed the title”
End Signal
```

The event follows the form on_..._triggered(Checked As Boolean), where ... stands for the object name set in Qt Designer for that button.

It creates an event procedure used for Qt Designer objects, which are executed then the event is triggered (in our case when the button is pressed).

Do you need to tell Qt Designer about your created Outlet and Signal?

We are nearly finished with your example, the last task we need to do is to connect the Outlet written in the source code file with the button created in the window.

Truly, you do not need to connect it or tell Qt Designer about your code changes, because it is automatically done for you. You only thing you need to keep in mind, is to name your objects meaningful (set the the object name of every controls).

In the IDE of Q7Basic select **build & run your example**. After the first message is shown, press on the button appearing on screen and the title of the button is changed after clicking.

To be continued...

Copyright