Q7 Tutorial

Version of 18 April 2012

A Q7Basic course

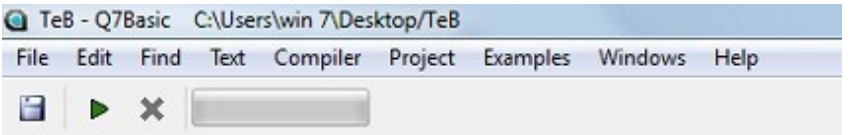
Introduction

Computer programming involves the use of different languages to give instructions to perform a certain task. One of the new developments in computer languages is Q7Basic. What characterisis Q7Basic? Q7Basic uses windows also called Forms. Q7Basic programs are event-driven. It means the program's execution depends on the actions taken by the user, such as clicking the mouse or using the keyboard. The notion “event-driven” program development has become part of the information vocabulary. The main focus of this tutorial is to teach how to develop “event-driven” programs. It is intended to be used together with a computer equipped with “Q7Basic”,[1] not only theoretical but also practical abilities have to be developed. First we determine what actions the user is allowed to take. Once the application is running the user sees only the form as a window.

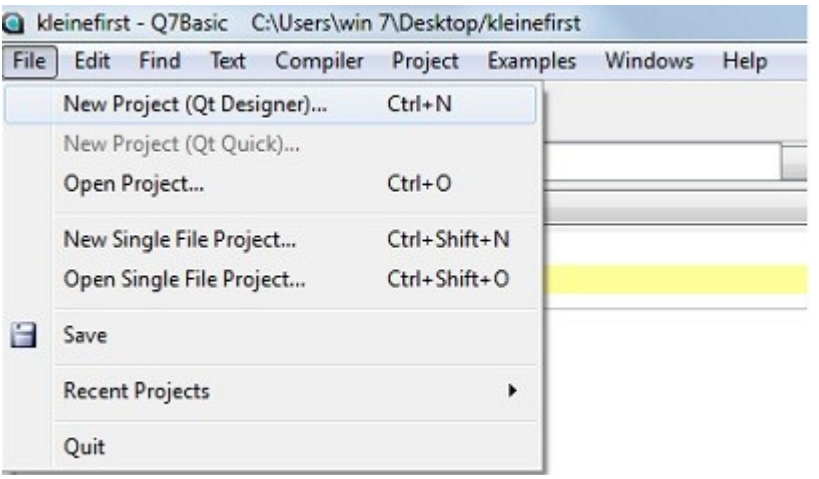
Starting Q7Basic

Using the trial option, Q7Basic can be used without any monetary obligation. However, to continue would mean to register with a minimal amount to pay. If a user decides to upgrade to a professional version which offers more features, then it would be a bit expensive. For beginners the professional features are definitely not necessary.

To start Q7Basic click the “trial”-button to open the window shown below. The upper left part is shown.



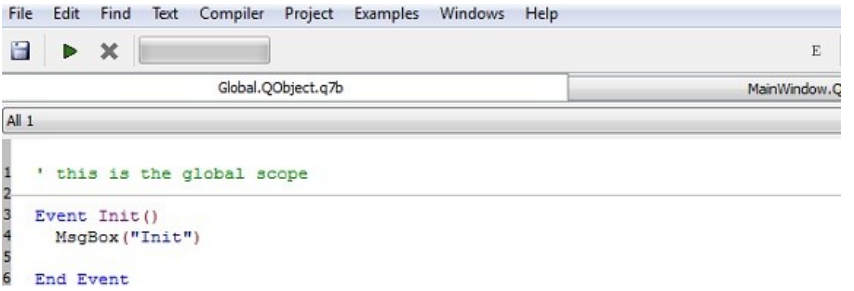
Choose from “File” the item “New Project (Qt Designer)” or type Ctrl+N as shown below.



A small window opens asking for the name of the new project. See

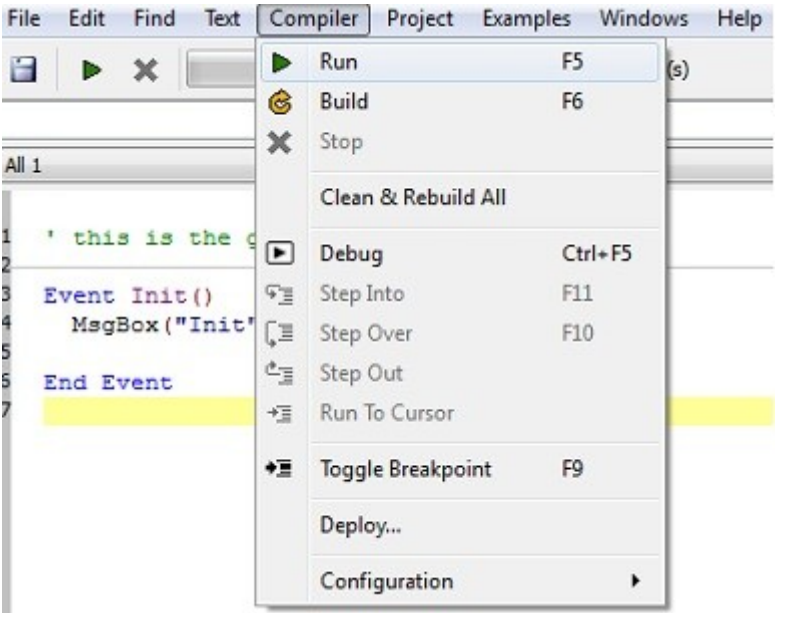
below:

After having given a name to the project click OK and the next window opens:



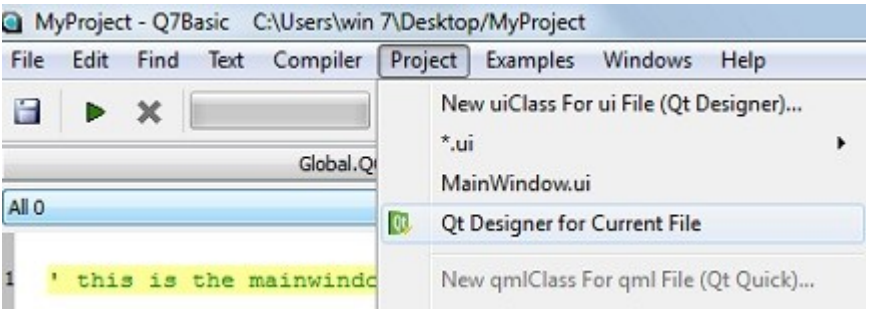
It shows the window “Global.QObject.q7b”. The “MainWindow....” will contain the subroutines handling the form-events.

The Global window contains some text and program code. To see whether Q7basic is funcioning correctly execute the code by clicking “run” under “Compiler” or F5. See the picture below:



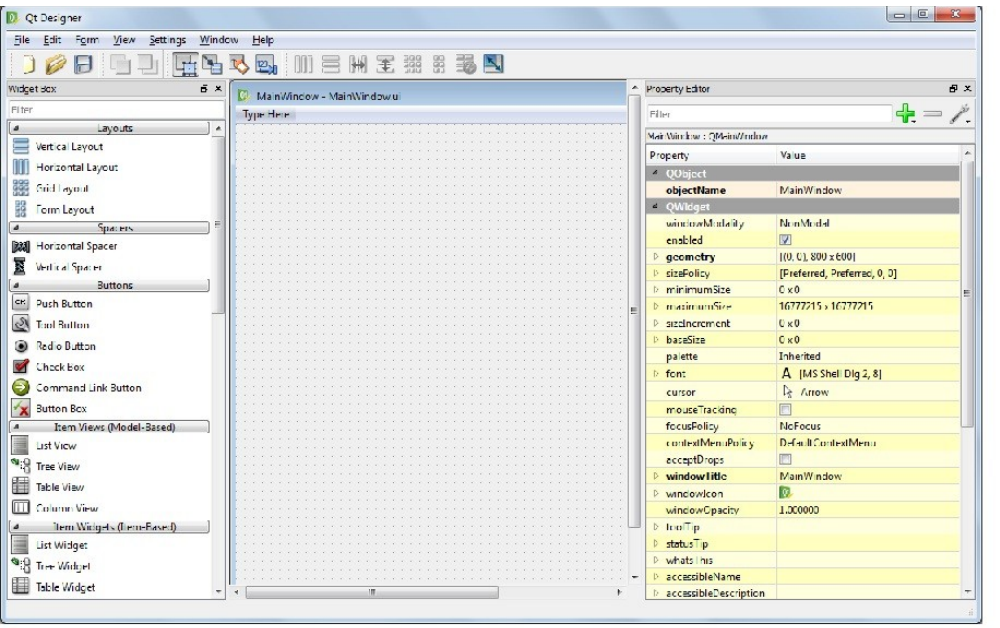
A message box appears with the text"Init". Clicking "OK" the Main window of the application opens but is empty because program code is not inserted yet. Hence, close it. Note that an application can also be closed by clicking the menu item "stop".

As an exercise, replace the text with something and run the program again.



Put a "push"-button on the form of this application. To do it start the Qt designer. See the picture below. Be sure that you first click "Main Window....", (It is the one right of "Global") otherwise an error message will be the result.

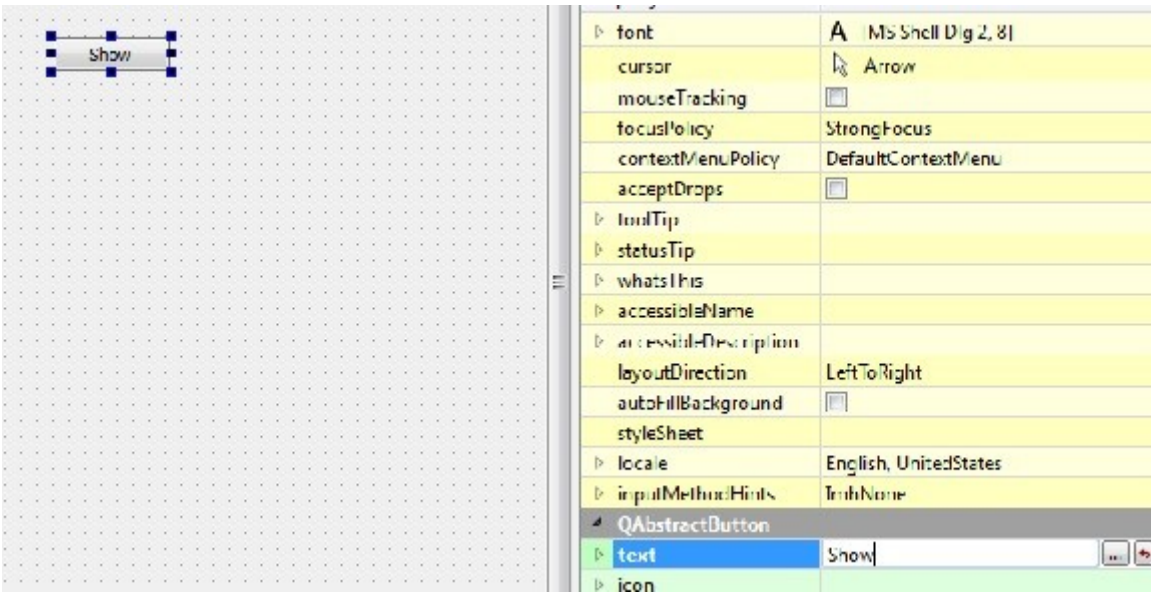
The window with the form opens:



Left of the form a window called "Widget Box" is shown and to the right is the "Property Editor". If they are invisible choose the menu item "View" and click their names which are called "widgets". In other Basic verions they are called "controls".To move a button from the widget box to the form drag it keeping the left mouse button pressed.

Below is the result:

In the Property Editor the object name is pushButton. Change it to “showbtn”. Scroll down the Property Editor until its green part appears. Change the “Text” text into Show. The result is shown immediately:



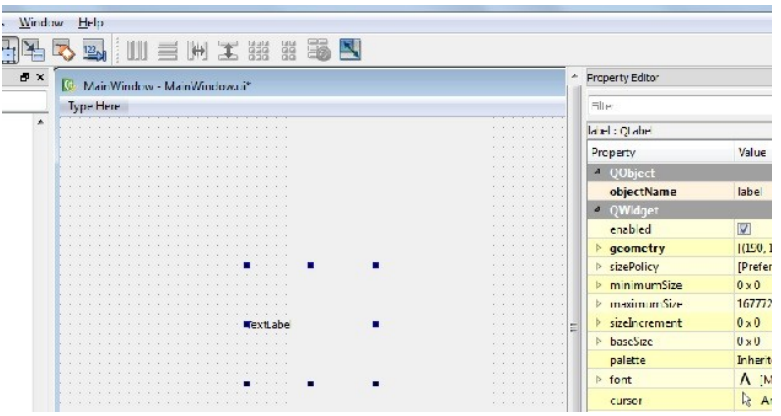
Then in the menu “File“ click the item “Save” and then “Quit” to return to th Q7B editor. (The Main window is still open). The following code has to be inserted:

```
Signal on_showbtn_clicked(Checked As Boolean)
  MsgBox("Show")
End Signal
```

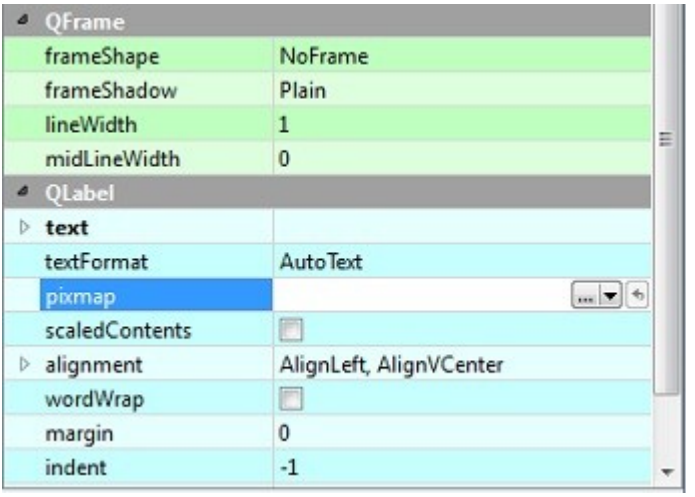
Make sure that “End Signal” is at the same position as “Signal on.....” and that “showbtn” is exactly the same as in the Qt designer (case sensitive)! Once both message boxes are shown, the one in the “global” section can be deleted.

A Picture

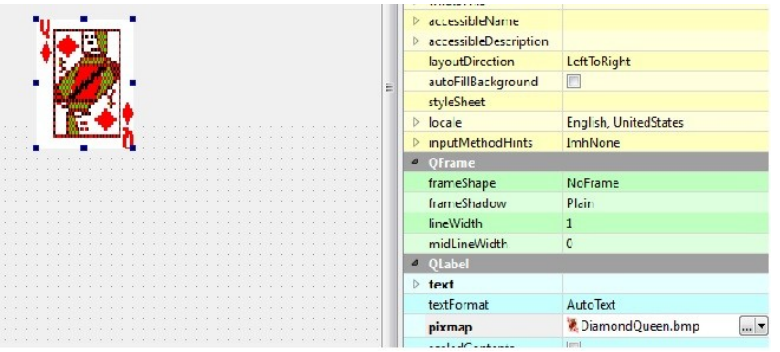
The next application will show a picture. Because “ImageBox”has not yet been implemented use the control “Label” for that purpose and use a bitmap (bmp) file. Files with the “jpg” extension are not accepted. Proceed by opening a new project, give it a name and open the Qt designer to place a label on the form, as the following picture shows. It has been enlarged already.



The text has to be removed before inserting the picture. Move to the green part of the Property Editor and click the little arrow of “pixmap”. After selecting it the item becomes blue:

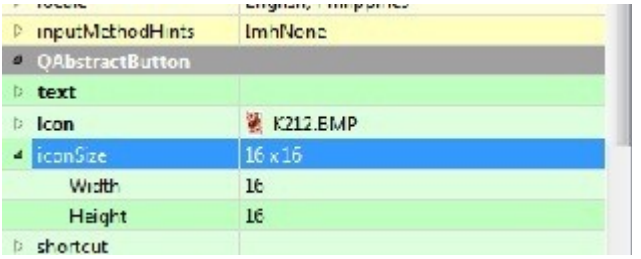


In the above example, the Diamond Queen was chosen. See the result below after the label had been adjusted to the picture:

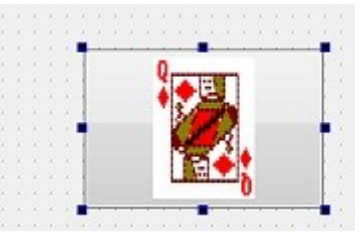


Save the form, close the Qt designer, click “run” or F5.
No code needed!

Now will be disclosed how to handle the Queen as push button.
Start a new project, for example “Diamond queen” and on the form place a button called DiamondQueenbtn.
To adapt the size of the button scroll down in the property window until “text” appears. See below:



Remove the text and then click on the small triangle left of “IconSize” to adjust “Width” and “Height”. For example 1600 for both. It is immediately reflected on the form:



And can be adjusted:

After saving quit the Qt designer and insert the following code:

```
Outlet DiamondQueenbtn As QPushButton
  Signal on_DiamondQueenbtn_clicked(Checked As Boolean)
    DiamondQueenbtn.Visible = False
  End Signal
```

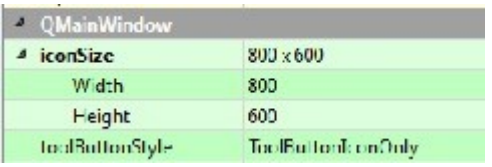
Add a control to the form to bring the Queen back on the screen:

```
Signal on_Showbtn_clicked(Checked As Boolean)
  DiamondQueenbtn.Visible = True
End Signal
```

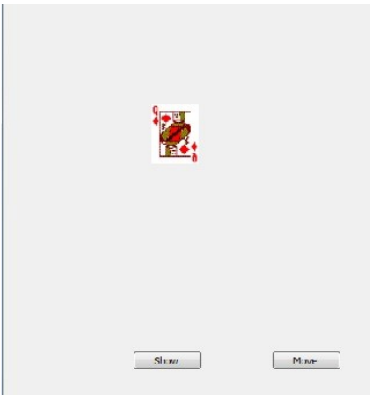
Then add another widget (control) to move the card:

```
Signal on_Movebtn_clicked(Checked As Boolean)
DiamondQueenbtn.x = 40
DiamondQueenbtn.y = 20
End Signal
```

With the Qt designer adjust the picture of the Queen and adjust too its values with the project editor as shown below:



The following picture shows the “diamond queen” screen.

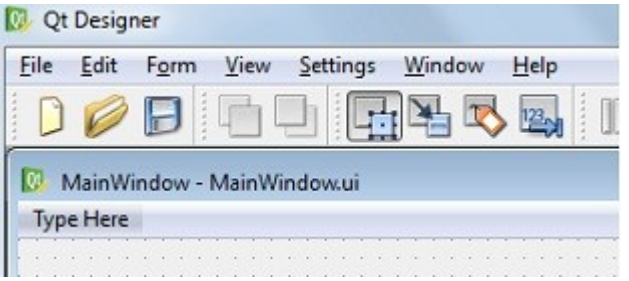


See for the complete program the listings: diamond queen

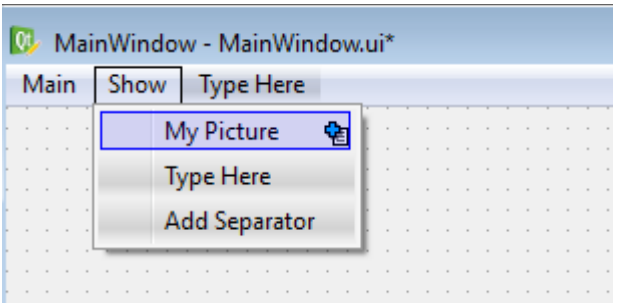
The Menu

Window users are accustomed to use menus. A well-designed application should have a menu. To include a menu in your application take the following steps:

1.Start a new project and give a name like “HappyMenu”
2.Start the Qt Designer
3.Type the menu at “Type Here”.
See below:

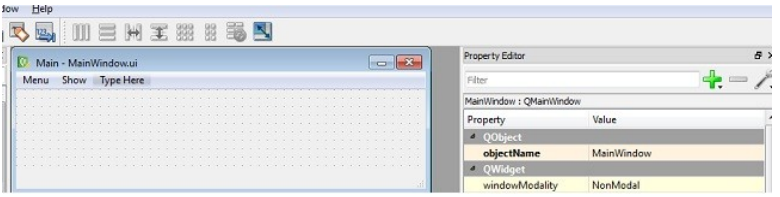


For example:



4. Save and then close the designer

Below a Qt Designer picture is provided to see if your form is the same as pictured below.



5. Insert the code in the Main window editor as follows:

```
' this is the mainwindow scope
Signal on_actionExit_triggered(Checked As Boolean)
  Application.Quit()
End Signal
'
Signal on_actionMy_Picture_triggered(Checked As Boolean)
    MsgBox("This is my text!!")
End Signal
```

6. Run the program to see whether it functions.

(The text about inserting a menu has been suggested by J. van Zijl)

As an exercise add a menu item and display a picture.

File manipulation

Next an application which shows an address. By developing it learn about a few other control items. To begin with, what should appear on the screen is as follows

Emilie Sagario

Adamville Compound

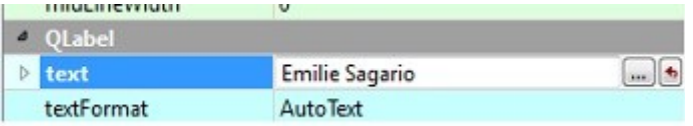
Marigondon, LapuLapuCity

0963848889

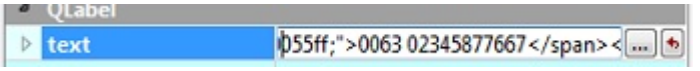
To implement it in Q7Basic create a new project and after selecting "MainWindow....." select the Qt designer. The "widget" (control) label will be used to show the different items. As shown below it can be found under "Display Widgets".

Drag it to the form and change its properties. The first one is its name:

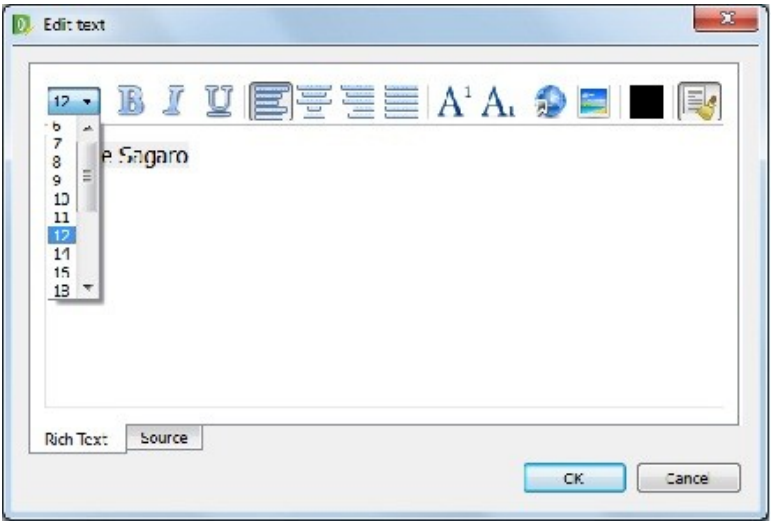
It is “Namelbl”. Some other entries have to be changed too:



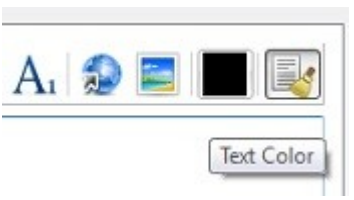
The text has been changed to what is to be displayed. How it will be displayed will be adapted too. Clicking the three dots a form to do it appears. The font size has been adapted as shown below. It is wise to adapt the label to how it should be displayed if you want all labels displayed in the same way because then you can simply copy and paste the item and then change for example the text. Be sure to change the correct part of the string: (here the telephone number)



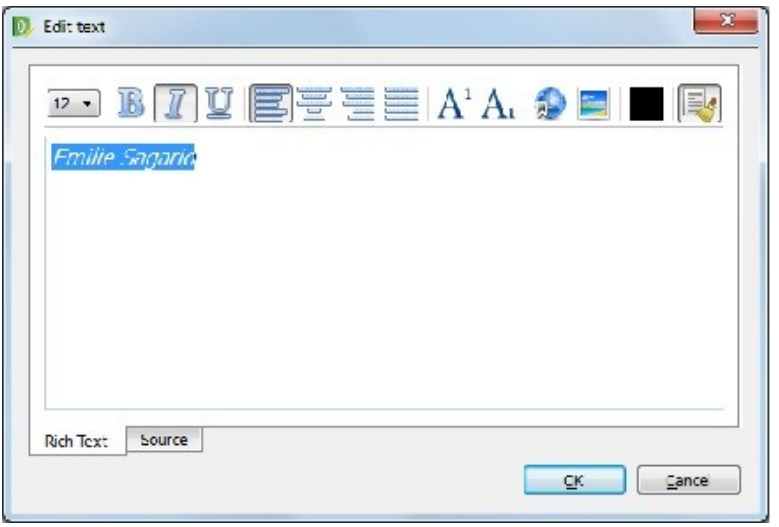
Below is shown the window that enables to edit, the font size 12 is chosen:



Also the color can be adapted by clicking the black square:

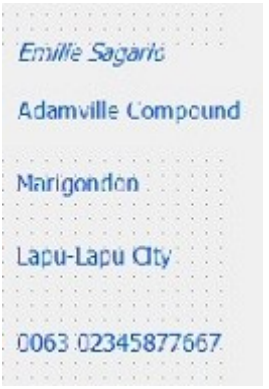


A window for editing opens:



After choosing a color the label is shown as:

and after inserting all the labels the result is:

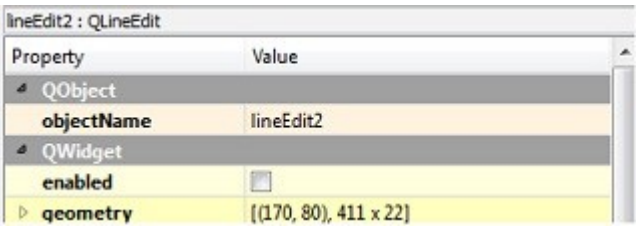


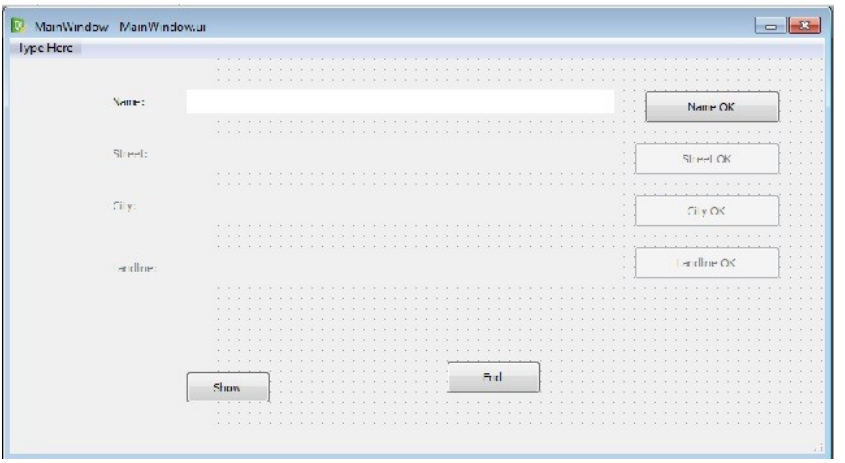
Now develop an application that allows to enter the data which will appear on the screen.

First, design on paper the desired window:

When the design is ready we implement it in Q7Basic by creating a new project , selecting “MainWindow.....” and selecting the Qt designer. The “widget' (control) label will be used to show what is expected and also an input label to enter text.

The form should be as shown on the next page. Note that most items are not enabled, for example “lineEdit2”:





After the form is ready and saved the code has to be inserted in the Main Window. After the dim-statements for the variables come the statements for the controls:

```
Outlet pushButton1 As QpushButton
Outlet pushButton2 As QPushButton
Outlet pushButton3 As QPushButton
Outlet pushButton4 As QPushButton
Outlet Show As QPushButton
Outlet EndProgram As QPushButton
Outlet lineEdit1 As QLineEdit
Outlet lineEdit2 As QLineEdit
Outlet lineEdit3 As QLineEdit
Outlet lineEdit4 As QLineEdit
Outlet Streetlbl As QLabel
Outlet Citylbl As QLabel
```

```
Outlet Landlinelbl As QLabel
```

Then the different controls follow. Note that only the first line of the form is visible. (The others when text has been entered and the OK button for that line has been pressed). As an example the code for Button1 follows:

```
Signal on_pushButton1_clicked(Checked As Boolean)
Naamstr = lineEdit1.Text
lineEdit2.enabled = True
pushButton2.enabled = True
Streetlbl.enabled = True
End Signal
```

Still it is not pleasant for the users to have to click many times the Message Box, therefore the result is depicted in a separate window. To do so four labels are added:

```
Showlbl1.Text = Naamstr
Showlbl2.Text = Straatstr
Showlbl3.Text = Citystr
Showlbl4.Text = Landlinestr
```

and made known to the compiler by inserting:

```
Outlet Showlbl1 As Qlabel
Outlet Showlbl2 As QLabel
Outlet Showlbl3 As QLabel
Outlet Showlbl4 As QLabel
```

Now when clicking the approriate button the above items are displayed.

The data entered should be stored in a file. Only the routine “Show” needs to be adapted:

```
Signal on_Show_clicked(Checked As Boolean)
        s = Naamstr & "\r\n"
        s = s & Straatstr &  "\r\n"
        s = s & Citystr &  "\r\n"
        s = s & Landlinestr & "\r\n"
        WriteString(s, "D:/text.txt")
End Signal
```

The addition "\r\n" ensures that each item appears on a new line when printed. If you want to appear everything in one line another delimiter has to be chosen. As the character “#” is seldom used in an address it could be used.

To read the file created insert the following code:

```
Signal on_Read_clicked(Checked As
Boolean)
        ReadString(s, "D:/test.txt")
End Signal
```

It s clear from this code that a button had been added called “Read”.

A more elaborate application has been provided by J. Van Zijl. It enables reading, writing and editing addresses. The first time the program runs the following message is shown:

After clicking OK the next window opens:

After filling in the address data the long blue button has to be pressed. The "Add contact" button becomes available. After clicking it the data are stored and the next address can be typed. When a few addresses are input they can be viewed by scrolling the long blue button.

The complete code can be found in the listings under "adres3".

Listings:

adres (version 1)

```
'
' Slowdown September 12 2010
'adapted by dr f j meijer


' this is the mainwindow scope
Dim Naamstr As String
Dim Straatstr As String
Dim Citystr As String
Dim Landlinestr As String
'
Outlet pushButton1 As QPushButton
Outlet pushButton2 As QPushButton
Outlet pushButton3 As QPushButton
Outlet pushButton4 As QPushButton
Outlet Startbtn As QPushButton
Outlet Show As QPushButton
Outlet EndProgram As QPushButton
Outlet lineEdit1 As QLineEdit
Outlet lineEdit2 As QLineEdit
Outlet lineEdit3 As QLineEdit
Outlet lineEdit4 As QLineEdit
Outlet Streetlbl As QLabel
Outlet Citylbl As QLabel
Outlet Landlinelbl As QLabel
Outlet Showlbl1 As QLabel
Outlet Showlbl2 As QLabel
Outlet Showlbl3 As QLabel
```

```
Outlet Showlbl4 As Qlabel
'
Signal on_pushButton1_clicked(Checked As Boolean)
Dim NewText As String
Naamstr = lineEdit1.Text
lineEdit2.enabled = True
pushButton2.enabled = True
Streetlbl.enabled = True
End Signal
'
Signal on_pushButton2_clicked(Checked As Boolean)
Dim NewText As String
Straatstr = lineEdit2.Text
lineEdit3.enabled = True
pushButton3.enabled = True
Citylbl.enabled = True
End Signal
'
Signal on_pushButton3_clicked(Checked As Boolean)
Dim NewText As String
Citystr = lineEdit3.Text
lineEdit4.enabled = True
pushButton4.enabled = True
Landlinelbl.enabled = True
End Signal
'
Signal on_pushButton4_clicked(Checked As Boolean)
Dim NewText As String
Landlinestr = lineEdit4.Text
End Signal
'
Signal on_Show_clicked(Checked As Boolean)
MsgBox(Naamstr)
MsgBox(Straatstr)
MsgBox(Citystr)
MsgBox(Landlinestr)
End Signal
'
Signal on_Startbtn_clicked(Checked As Boolean)
Showlbl1.Text = Naamstr
Showlbl2.Text = Straatstr
Showlbl3.Text = Citystr
Showlbl4.Text = Landlinestr
End Signal
'
Signal on_EndProgram_clicked(Checked As Boolean)
Application.Quit()
End Signal
```

_ _

adres (version 2)

```
' Slowdown September 12 2010
'adapted by dr f j meijer
'
' this is the mainwindow scope
Dim Naamstr As String
Dim Straatstr As String
Dim Citystr As String
Dim Landlinestr As String
Dim s As String
'
Outlet pushButton1 As QPushButton
Outlet pushButton2 As QPushButton
Outlet pushButton3 As QPushButton
Outlet pushButton4 As QPushButton
Outlet Startbtn As QPushButton
Outlet Show As QPushButton
Outlet EndProgram As QPushButton
Outlet lineEdit1 As QLineEdit
Outlet lineEdit2 As QLineEdit
Outlet lineEdit3 As QLineEdit
```

```
Outlet lineEdit4 As QLineEdit
Outlet Streetlbl As QLabel
Outlet Citylbl As QLabel
Outlet Landlinelbl As QLabel
Outlet Showlbl1 As QLabel
Outlet Showlbl2 As QLabel
Outlet Showlbl3 As QLabel
Outlet Showlbl4 As QLabel
'
Signal on_pushButton1_clicked(Checked As Boolean)
Dim NewText As String
Naamstr = lineEdit1.Text
lineEdit2.enabled = True
pushButton2.enabled = True
Streetlbl.enabled = True
End Signal
'
Signal on_pushButton2_clicked(Checked As Boolean)
Dim NewText As String
Straatstr = lineEdit2.Text
lineEdit3.enabled = True
pushButton3.enabled = True
Citylbl.enabled = True
End Signal
'
Signal on_pushButton3_clicked(Checked As Boolean)
Dim NewText As String
Citystr = lineEdit3.Text
lineEdit4.enabled = True
pushButton4.enabled = True
Landlinelbl.enabled = True
End Signal
'
Signal on_pushButton4_clicked(Checked As Boolean)
Dim NewText As String
Landlinestr = lineEdit4.Text
End Signal
'
Signal on_Show_clicked(Checked As Boolean)
s = Naamstr & "#"
s = s & Straatstr & "#"
s = s & Citystr & "#"
s = s & Landlinestr & "#"
WriteString(s, "D:/tet.txt")
End Signal
'
Signal on_Read_clicked(Checked As Boolean)
ReadString(s, "D:/test.txt")
End Signal
'
Signal on_Startbtn_clicked(Checked As Boolean)
Showlbl1.Text = Naamstr
Showlbl2.Text = Straatstr
Showlbl3.Text = Citystr
Showlbl4.Text = Landlinestr
End Signal
'

Signal on_EndProgram_clicked(Checked As Boolean)
Application.Quit()
End Signal
```

_ _

diamond queen

```
' this is the mainwindow scope
Outlet DiamondQueenbtn As QPushButton
Outlet Showbtn As QPushButton
Outlet Movebtn As QPushButton
'
Dim x As Integer
Dim y As Integer
```

```
Signal on_DiamondQueenbtn_clicked(Checked As Boolean)
DiamondQueenbtn.Visible = False
End Signal
'
Signal on_Showbtn_clicked(Checked As Boolean)
DiamondQueenbtn.Visible = True
End Signal
'
Signal on_Movebtn_clicked(Checked As Boolean)
DiamondQueenbtn.x = 40
DiamondQueenbtn.y = 20
End Signal
```

_ _

adres3

globals:

```
' this is the global scope
Public Dim NameFile As String = Application.Path & "/Adress_Name.xml"
Public Dim StreetFile As String = Application.Path & "/Adress_Street.xml"
Public Dim CityFile As String = Application.Path & "/Adress_City.xml"
Public Dim TelephoneFile As String = Application.Path & "/Adress_Telephone.xml"
```

main:

```
' April 15 2012
' Simple example about getting, writing and re-writing data.
' You can find me on,
' http://www.q7basic.org/forum/index.php and
' http://www.kbasic.com/forum/index.php
' j.vanzijl@quicknet.nl


' this is the mainwindow scope
Outlet AdresscomboBox As QComboBox
Outlet AdNewpushButton As QPushButton
Outlet RemovepushButton As QPushButton
Outlet EndpushButton As QPushButton


Outlet NamelineEdit As QLineEdit
Outlet StreetlineEdit As QLineEdit
Outlet CitylineEdit As QLineEdit
Outlet LandlinelineEdit As QLineEdit



Dim Name As List
Dim Street As List
Dim City As List
Dim Telephone As List
Dim CurrentIndex As Integer


Event Init()
AdNewpushButton.Enabled = False
RemovepushButton.Enabled = False
LoadAdressData()
SetFocus(NamelineEdit)' setfocus to Qwidget, in this case set focus to NamelineEdit
End Event


Sub LoadAdressData()
Dim AllPressent As Boolean
Dim Lus As Integer


AllPressent = True


If FilePressent(NameFile) = True Then
Name = ReadList(NameFile)
Else
AllPressent = False
```

```
End If
If FilePressent(StreetFile) = True Then
Street = ReadList(StreetFile)
Else
AllPressent = False
End If
If FilePressent(CityFile) = True Then
City = ReadList(CityFile)
Else
AllPressent = False
End If
If FilePressent(TelephoneFile) = True Then
Telephone = ReadList(TelephoneFile)
Else
AllPressent = False
End If


If AllPressent = False Then
' when no data is found alert user.
MsgBox("Warning ", "No Adress data found")
Else
' data is found so we load the names into a combobox.
AdresscomboBox.RemoveAll()
For Lus = 0 To Name.Length() - 1
AdresscomboBox.Append(Name.Object(Lus))
Next
End If
End Sub


Signal on_NamelineEdit_editingFinished()
SetFocus(StreetlineEdit)
End Signal


Signal on_LandlinelineEdit_editingFinished()
If Len(NamelineEdit.Text) > 0 And Len(StreetlineEdit.Text) > 0 And Len(CitylineEdit.Text) > 0 Then
AdNewpushButton.Enabled = True
SetFocus(AdNewpushButton)
End If
End Signal


Signal on_StreetlineEdit_editingFinished()
SetFocus(CitylineEdit)
End Signal


Signal on_CitylineEdit_editingFinished()
SetFocus(LandlinelineEdit)
End Signal


Signal on_AdNewpushButton_pressed()
Dim IsWritten As Boolean
Dim AllWritten As Boolean


AllWritten = True
Name.Append(NamelineEdit.Text)
Street.Append(StreetlineEdit.Text)
City.Append(CitylineEdit.Text)
Telephone.Append(LandlinelineEdit.Text)


IsWritten = WriteList(Name, NameFile)
If IsWritten = False Then
AllWritten = False
End If
IsWritten = WriteList(Street, StreetFile)
If IsWritten = False Then
AllWritten = False
End If
IsWritten = WriteList(City, CityFile)
If IsWritten = False Then
```

```
AllWritten = False
End If
IsWritten = WriteList(Telephone, TelephoneFile)
If IsWritten = False Then
AllWritten = False
End If
If AllWritten = True Then
LoadAdressData()
AdNewpushButton.Enabled = False
NamelineEdit.Text = ""
StreetlineEdit.Text = ""
CitylineEdit.Text = ""
LandlinelineEdit.Text = ""
SetFocus(NamelineEdit)
Else
MsgBox("Warning ", "Some data isn't correctly written to disk")
End If
End Signal


Signal on_AdresscomboBox_activated(Text As String)
CurrentIndex = AdresscomboBox.CurrentIndex
NamelineEdit.Text = Name.Object(CurrentIndex)
StreetlineEdit.Text = Street.Object(CurrentIndex)
CitylineEdit.Text = City.Object(CurrentIndex)
LandlinelineEdit.Text = Telephone.Object(CurrentIndex)
RemovepushButton.Enabled = True
End Signal


Signal on_RemovepushButton_pressed()
Dim IsWritten As Boolean
Dim AllWritten As Boolean


AllWritten = True


RemovepushButton.Enabled = False
Name.Remove(CurrentIndex)
Street.Remove(CurrentIndex)
City.Remove(CurrentIndex)
Telephone.Remove(CurrentIndex)
IsWritten = WriteList(Name, NameFile)
If IsWritten = False Then
AllWritten = False
End If
IsWritten = WriteList(Street, StreetFile)
If IsWritten = False Then
AllWritten = False
End If
IsWritten = WriteList(City, CityFile)
If IsWritten = False Then
AllWritten = False
End If
If AllWritten = True Then
LoadAdressData()
AdNewpushButton.Enabled = False
NamelineEdit.Text = ""
StreetlineEdit.Text = ""
CitylineEdit.Text = ""
LandlinelineEdit.Text = ""
SetFocus(NamelineEdit)
Else
MsgBox("Warning ", "Some data isn't correctly written to disk")
End If
End Signal


Public Function FilePressent(MyFile As String) As Boolean
' Check if file exists and is bigger then zero bytes.
If File.Size(MyFile) > 0 Then
Return True
Else
Return False
```

```
End If
End Function

Signal on_EndpushButton_clicked(Checked As Boolean)
Application.Quit()
End Signal
```