

Toolbar for VisualBasic (Version 1.1a)  
(c) 1993-1994 Andreas Fehr, Switzerland  
CIS 100042.2070

Your Project has already

- MDIParent form
- central \*.bas file (with your important code)

Add the following files to your project

- FILE5.frm (wndResources)
- FILE7.frm (wndToolbar)
- FILE6.frm (wndQHelp)
- FILE4.bas
- FILE3.bas
- FILE1.bas

**What you have to do is:**

**a. Easy part**

(1)

Add two PictureBoxes (BorderStyle = 0) to your MDIParent

- one Align=Top name Toolbar
- one Align=Bottom name Statusbar

(2)

Add a label to Statusbar and name it status.

(AutoSize = True)

(3)

Add the following line to the MouseMove event of the Toolbar:

Call vbQHelpExt(Toolbar, bool)

(where bool is an integer (True/False) whether you want QuickHelp to use or not)

(4)

Add the following line to the Resize events of the Toolbar / Statusbar

MakeStatusBar Toolbar

MakeStatusBar Statusbar

(5)

Load wndResources and copy your Tool-Bitmaps to the array of Images.

Add a StatusText +";"+MouseText to the tag property of the images:

e.g. Open an existing document;Open.

(6)

Copy the code of the sub ToolCalled (FILE2.BAS) to your central .bas file.

(7)

Copy the code of the sub ToolCopy (FILE2.BAS) to your central .bas file.

(8)

Add the following line to the MDIParent\_Unload event:

vbQHExit Me

**b. Hacking part (looks like some work, but it's still easy):**

(9)

Change the code in ToolCalled to fit your project:

```
Sub ToolCalled (nr As Integer, Status As Label)
    Select Case nr      ' Select Case on your toolnumber
        Case 0          ' put the appropriate code to the
        Case 1          ' according toolnumber
        Case 2
        Case ...
    End Select
End Sub
```

This wasn't that hard, was it?

(10)

Add the following line to your central .bas file

Global Const ToolCnt = x

whereas x is the amount of tools you have, e.g.

Global Const ToolCnt = 10

This wasn't hard either, was it?

(11)

Add a (checked) menu to your MDIParent to turn on/off the toolbar

e.g. MenuO\_Sho (Menu.Options.ShowToolbar)

and place the following code in the Click event:

```
If MenuO_Sho.Checked Then
    Toolbar.Visible = False
Else
    vbQHShowTool
End If
MenuO_Sho.Checked = Not MenuO_Sho.Checked
```

(12)

Change to code in ToolCopy to fit your project:

```
' Target = top aligned PictureBox on MDIParent, called Toolbar
Sub ToolCopy (Target As PictureBox)
    Dim i As Integer
    Dim temp1$, temp2$
    Dim TempTool As ToolType
    vbQHUsed True
    ' ToolCnt      (defined in point 9; do not change)
    ' wndToolbar   (form for floating toolbar; do not change)
    ' Target       (top aligned PictureBox of MDIParent; called Toolbar)
    '              (menu to choose whether to show the Toolbar or not)
    ' Target.Parent.MenuO_Sho
    '              (label to print the current status on)
    ' Target.Parent.Status
    ' "Ready"      (default message to print on status)
    ' Put the following statement on one line
    Call vbInitTools (ToolCnt, wndToolbar, Target,
        Target.Parent.MenuO_Sho,
        Target.Parent.Status, "Ready")
```

```

' load form with resources
Load wndResources

' Copy the first tool
i = InStr(wndResources.Tool(0).Tag, ";")
If i Then
    temp1$ = Left$(wndResources.Tool(0).Tag, i - 1)
    temp2$ = Right$(wndResources.Tool(0).Tag,
Len(wndResources.Tool(0).Tag) - i)
Else
    temp1$ = wndResources.Tool(0).Tag
End If

TempTool.Group = 1
TempTool.Visible = True
TempTool.Enabled = True
TempTool.qHelp = True
TempTool.StatText = temp1$
TempTool.MouseText = temp2$
TempTool.UseMouse = True
TempTool.CopyPicture = True
vbCopyToolExt TempTool, 0, wndResources.Tool(0), Target

' Copy the second tool
i = InStr(wndResources.Tool(1).Tag, ";")
If i Then
    temp1$ = Left$(wndResources.Tool(1).Tag, i - 1)
    temp2$ = Right$(wndResources.Tool(1).Tag,
Len(wndResources.Tool(1).Tag) - i)
Else
    temp1$ = wndResources.Tool(1).Tag
End If
TempTool.Group = 1
TempTool.Visible = True
TempTool.Enabled = True
TempTool.qHelp = True
TempTool.StatText = temp1$
TempTool.MouseText = temp2$
TempTool.UseMouse = True
TempTool.CopyPicture = True
vbCopyToolExt TempTool, 1, wndResources.Tool(1), Target

' Copy the third tool etc...

' unload resources
Unload wndResources
End Sub

```

This is some more work, but it's all done now.

The interface (alphabetical):

```

vbQHCOPYToolExt tTool, nr, Source, Target
    Call once for each tool during starting procedure.
        tTool    ToolType Structure
        nr        integer, Toolnumber

```

source PictureBox with original picture  
target Toolbar (=PictureBox) where to copy tools  
Called only once!

vbQHelpExt Toolbar

Basic function of toolbar. Call this in the Toolbar\_MouseMove event.  
Toolbar PictureBox  
Called a hundred time (Everytime the mouse moves over toolbar).  
(Called "automatically")

vbQHEnabled nr, flag

Called to enable / disable  
nr integer (Tool to enable/disable)  
flag integer (True/False)  
Called to change status of tool. Call explicit to change it.

vbQHExit Me

Called in the MDIParent\_Unload event.  
Me MDIParentform  
Called once to free resources. (Called "automatically")

vbQHInitTools cnt, tb, Target, mnu, lbl, cap

Called once to initialise the resources used.  
cnt integer, cnt of tools  
tb Form (Floating toolbar)  
mnu Menu to uncheck if floating toolbar unloads  
lbl Label where to write some status informations  
cap String with caption for default status information  
Called once for each tool you want to load on your toolbar.

vbQHShowTool

Called everytime the floating toolbar has unloaded.

vbQHUsed flag

Called every time you'd like to turn qHelp on or off.  
flag integer (True/False)

MakeStatusBar pb

Placed in the PictureBox\_Resize event.  
pb PictureBox (that has been resized)  
(Called "automatically")

Called "automatically" means you have to place the code once and forget it.

If you still have some problems, send me a note:

CIS 100042.2070  
Andreas Fehr, Switzerland

Thank you and good luck