

MODULE 26: SYSTEM TOOLS

On Completion of this module you will be able to edit and manipulate system tools and add advanced solutions to elements.

Subject Outcome 1: Introduction

Subject Outcome 2: Accessories:

- 2.1 Sound Card Info
- 2.2 Reading/Writing Com Port (see [MS3 Networking](#))
- 2.3 (See 26.4.2 –keyboard status reading)
- 2.4 Sound monitor (Peak value monitor)

Subject Outcome 3: Software

- 3.1 Run Time Programs
- 3.2 Cursor Dialog Box
- 3.3 Prevent double program run – only one instance allowed
- 3.4 Alter Desktop Wallpaper

Subject Outcome 4: Element Enhancement

- 4.1 Tab Control Icon
- 4.2 Status bar images & reading of keyboard status
- 4.3 QRichEdit adding an image
- 4.4 SpellChecking QRichEdit
- 4.5 All image viewer – image dialog box
- 4.6 Tree Viewer (view info with tree module)



26.1 INTRODUCTION

The following are individual elements and applications to enhance your program. Some of them might be related to some of the modules, but due to its capability of manipulating the system, it is regarded as unique system enhancing tools.

26.2 ACCESSORIES

26.2.1 Sound Card Info

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON #include "rapidq.inc" Type WavelnCaps ManufacturerID As Short ProductID As Short DriverVersion As Long ProductName As String * 32 Formats As Long Channels As Short Reserved As Short End Type Declare Function wavelnGetNumDevs Lib "winmm" Alias "wavelnGetNumDevs" () As Long Declare Function wavelnGetDevCaps Lib "winmm.dll" Alias "wavelnGetDevCapsA" _ (uDeviceID As Long, lpCaps As WAVEINCAPS, uSize As Long) As Long Dim Caps As WavelnCaps Defint structSize structSize = SizeOf [Caps] wavelnGetDevCaps 0, Caps, StructSize CREATE Form AS QFORM Caption = "Form1: Structure Size:"+(Str\$(structSize)) Width = 400:Height = 240:Center CREATE Label1 AS QLABEL Caption = "NumDevs: " + str\$(wavelnGetNumDevs) Left = 0:Top = 0:Width = 312:Height = 20:Transparent = 1 END CREATE CREATE Label2 AS QLABEL Caption = "Manufacturer id: " + str\$(Caps.ManufacturerID) Left = 0:Top = 21:Width = 312:Height = 20:Transparent = 1 END CREATE CREATE Label3 AS QLABEL Caption = "ProductID: " + str\$(Caps.ProductID) Left = 0:Top = 42:Width = 312:Height = 20:Transparent = 1 END CREATE CREATE Label4 AS QLABEL Caption = "DriverVersion: " + str\$(Caps.DriverVersion) Left = 0:Top = 63:Width = 312:Height = 20:Transparent = 1 END CREATE CREATE Label5 AS QLABEL Caption = "ProductName(1 To 32): " + Caps.ProductName Left = 0:Top = 84:Width = 312:Height = 20:Transparent = 1 END CREATE END CREATE Form.ShowModal </pre>	<p>Note NO rapidq.inc</p> <p>Variable holders.</p> <p>Establish element</p> <p>Structuring the element</p>

26.2.2 Reading/writing Com Port



SNO	CODING	EXPLANATION
1	<pre> \$Aptype GUI \$TYPECHECK ON \$OPTIMIZE ON \$INCLUDE <RAPIDQ2.INC> DECLARE SUB OpenClick DECLARE SUB SendClick DECLARE SUB ReadClick DECLARE SUB ComError(strErrorMessage AS STRING) DECLARE SUB PortOpen DECLARE SUB PortClose DECLARE SUB PortClosed DECLARE SUB StringWritten DECLARE SUB StringRead DIM strPort AS STRING DIM dwBaud AS DWORD DIM bParity AS BYTE DIM bDataBits AS BYTE DIM bStopBits AS BYTE DIM MyPort AS QCOMPORT MyPort.OnComError = ComError MyPort.OnOpen = PortOpen MyPort.OnClose = PortClosed MyPort.OnWriteString = StringWritten MyPort.OnReadString = StringRead DIM Font AS QFont Font.Name="Courier New" CREATE Form AS QFORM Caption = "Form1" Width = 820 Height = 355 Center CREATE lblPort AS QLABEL Caption = "COM Port" Left = 10: Width = 100: Top = 5 END CREATE CREATE lblBaud AS QLABEL Caption = "Bits per second" Left = 10: Width = 100: Top = 30 END CREATE CREATE lblParity AS QLABEL Caption = "Parity" Left = 10: Width = 100: Top = 55 END CREATE CREATE lblDataBits AS QLABEL Caption = "Data bits" Left = 10: Width = 100: Top = 80 END CREATE CREATE lblStopBits AS QLABEL Caption = "Stop Bits" Left = 10: Width = 100: Top = 105 END CREATE CREATE lblSend AS QLABEL Caption = "String to Send":Left = 10: Width = 100: Top = 130 END CREATE CREATE cbPort AS QCOMBOBOX AddItems "COM1","COM2","COM3","COM4" Left = 120: Width = 70: Top = 5:ItemIndex = 0 END CREATE CREATE cbBaud AS QCOMBOBOX AddItems "9600","14400","19200","38400","56000","57600","115200" Left = 120: Width = 70: Top = 30:ItemIndex = 6 </pre>	



SNO	CODING	EXPLANATION
	<pre> END CREATE CREATE cbParity AS QCOMBOBOX AddItems "None", "Odd", "Even", "Mark", "Space" Left = 120: Width = 70: Top = 55 ItemIndex = 0 END CREATE CREATE cbDataBits AS QCOMBOBOX AddItems "4", "5", "6", "7", "8" Left = 120: Width = 70: Top = 80 ItemIndex = 4 END CREATE CREATE cbStopBits AS QCOMBOBOX AddItems "1", "1.5", "2" Left = 120: Width = 70: Top = 105 ItemIndex = 0 END CREATE CREATE Edit1 AS QEDIT Text = "" Left = 120: Top = 130: Width = 200 END CREATE CREATE RichEdit1 AS QRICHEDIT Left = 3: Top = 160: Width = 795: Height = 130 PlainText = 1 ReadOnly = 1 ScrollBars = 3 WordWrap = 0 Font = Font END CREATE CREATE StatusBar AS QStatusBar Left = 10: Top = 300: Width = 795: Height = 20 SizeGrip = FALSE AddPanels "Panel 1", "Panel 2" Panel(0).Caption = "Not Connected" Panel(0).Width = 645 Panel(1).Caption = "" END CREATE CREATE OpenButton AS QBUTTON Caption = "&Open Port" Left = 330: Top = 5: Height = 20 OnClick = OpenClick END CREATE CREATE SendButton AS QBUTTON Caption = "&Send" Left = 330: Top = 130: Height = 20 Enabled = 0 OnClick = SendClick END CREATE CREATE ReadButton AS QBUTTON Caption = "&Read" Left = 440: Top = 130: Height = 20 Enabled = 0 OnClick = ReadClick END CREATE END CREATE SUB OpenClick SELECT CASE cbPort.ItemIndex CASE 0 strPort = "COM1" CASE 1 strPort = "COM2" CASE 2 strPort = "COM3" CASE 3 strPort = "COM4" END SELECT SELECT CASE cbBaud.ItemIndex CASE 0 dwBaud = 110 </pre>	



SNO	CODING	EXPLANATION
	<pre> CASE 1 dwBaud = 300 CASE 2 dwBaud = 600 CASE 3 dwBaud = 1200 CASE 4 dwBaud = 2400 CASE 5 dwBaud = 4800 CASE 6 dwBaud = 9600 CASE 7 dwBaud = 14400 CASE 8 dwBaud = 19200 CASE 9 dwBaud = 38400 CASE 10 dwBaud = 56000 CASE 11 dwBaud = 57600 CASE 12 dwBaud = 115200 END SELECT SELECT CASE cbParity.ItemIndex CASE 0 bParity = NOPARITY CASE 1 bParity = ODDPARITY CASE 2 bParity = EVENPARITY CASE 3 bParity = MARKPARITY CASE 4 bParity = SPACEPARITY END SELECT SELECT CASE cbDataBits.ItemIndex CASE 0 bDataBits = 4 CASE 1 bDataBits = 5 CASE 2 bDataBits = 6 CASE 3 bDataBits = 7 CASE 4 bDataBits = 8 END SELECT SELECT CASE cbStopBits.ItemIndex CASE 0 bStopBits = ONESTOPBIT CASE 1 bStopBits = ONE5STOPBITS CASE 2 bStopBits = TWOSTOPBITS END SELECT MyPort.Port = strPort MyPort.BaudRate = dwBaud MyPort.Parity = bParity MyPort.DataBits = bDataBits MyPort.StopBits = bStopBits MyPort.Open MyPort.Purgeln IF MyPort.Connected=TRUE THEN OpenButton.Enabled = 0 SendButton.Enabled = 1 ReadButton.Enabled = 1 END IF </pre>	



SNO	CODING	EXPLANATION
	<pre> END SUB SUB SendClick DIM BytesRead AS DWORD IF Edit1.Text="" THEN SHOWMESSAGE "Please enter String to Send" ELSE MyPort.WriteString(Edit1.Text + CHR\$(13) +CHR\$(10), 1000) Edit1.Text="" WHILE MyPort.BytesNotRead > 0 RichEdit1.Text=RichEdit1.Text & MyPort.ReadString(MyPort.BytesNotRead, 500) '& is addition operator.. BytesRead=MyPort.BytesNotRead + BytesRead StatusBar.Panel(1).Caption = "Number of Bytes Read " & STR\$(BytesRead) WEND PortClose END IF END SUB SUB ReadClick DIM BytesRead AS DWORD MyPort.Purgeln WHILE MyPort.BytesNotRead > 0 RichEdit1.Text=RichEdit1.Text & MyPort.ReadString(MyPort.BytesNotRead, 5) '& is addition operator.. BytesRead=MyPort.BytesNotRead + BytesRead StatusBar.Panel(1).Caption = "Number of Bytes Read " & STR\$(BytesRead) WEND END SUB SUB ComError (strErrorMessage AS STRING) SHOWMESSAGE strErrorMessage END SUB SUB PortOpen StatusBar.Panel(0).Caption = "Connected to " & MyPort.Port END SUB SUB PortClose MyPort.Close END SUB SUB PortClosed StatusBar.Panel(0).Caption = "Not Connected" OpenButton.Enabled = 1 SendButton.Enabled = 0 END SUB SUB StringWritten SHOWMESSAGE "String was written" END SUB SUB StringRead SHOWMESSAGE "String was read" END SUB Form.ShowModal </pre>	



26.24 Sound monitor (peak value monitoring)

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim blank as qstringlist declare sub callsoundemu CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480:Center onclick=callsoundemu END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal sub callsoundemu if fileexists("c:\ms3\~common\rctmr.fet")=1 then kill "c:\ms3\~common\rctmr.fet" if fileexists("c:\ms3\~common\st1race.fet")=1 then kill "c:\ms3\~common\st1race.fet" blank.clear blank.additems "race1" blank.additems "00.00:0000" blank.savetofile "c:\ms3\~common\rctmr.fet" shell "c:\ms3\mybis\lgstimer.exe" form.caption="Started: "+time\$ if fileexists("c:\ms3\~common\rctmr.fet")=1 then kill "c:\ms3\~common\rctmr.fet" if fileexists("c:\ms3\~common\st1race.fet")=1 then kill "c:\ms3\~common\st1race.fet" end sub </pre>	<p>You will require LGSTIMER.EXE with your program. The resource files are required for LGSTIMER to run [it will record a timer and once the sound has been activated, the timer would be zeroed, to indicate the sound that was activated].</p> <p>Also remember to create the C:\MS3\~COMMON folder</p> <p>Click on FORM, set sensitivity, click START and make a sound to activate program.</p>

26.3 SOFTWARE

26.3.1 Run Time (programs currently running)

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON CONST HKEY_CLASSES_ROOT = &H80000000 CONST HKEY_LOCAL_MACHINE = &H80000002 CONST ctab = CHR\$(9):CONST clf = CHR\$(10) DIM Registry AS QREGISTRY create SaveDialog AS QSaveDialog:end create declare sub exitme declare sub saveme DECLARE SUB Button1Click DECLARE SUB listele(adres AS STRING) CREATE Form AS QFORM CAPTION = "RUN TIME COMMANDS" Width = 620:Height = 300:borderstyle=4:Center create q as qpanel left=5:top=5:height=290:width=610 create button as qbutton top=230:left=280:caption = "exit":onclick=exitme end create create button1 as qbutton top=230:left=180:caption = "save":onclick=saveme end create CREATE richedit1 AS qrichedit Left = 10:Top = 10:height = 212:width = 580:scrollbars=3 END Create END CREATE END CREATE Button1Click Form.SHOWMODAL SUB Button1Click Registry.RootKey = HKEY_LOCAL_MACHINE richedit1.text="" </pre>	



SNO	CODING	EXPLANATION
	<pre> CALL listele("Run") CALL listele("RunOnce") CALL listele("RunOnceEx") CALL listele("RunServices") CALL listele("RunServicesOnce") END SUB SUB listele(adres AS STRING) DIM i AS SHORT, txt AS STRING, regitem AS STRING, adres2 AS STRING adres2="Software\Microsoft\Windows\CurrentVersion\" + adres WITH Registry .OpenKey(adres2, 0) txt="" FOR i = 0 TO .ValueItemCount - 1:regitem = .ValueItem(i) txt = txt + regitem + ctab + .readstring(regitem) + clf NEXT txt = STRING\$(60,"")+ CHR\$(10)+ adres + CHR\$(10) + STRING\$(60,"")+ CHR\$(10)+ txt richedit1.addstring txt .closekey END WITH END SUB sub exitme end end sub sub saveme if richedit1.linecount=0 then goto g53 savedialog.initialdir = "c:\ms3\~common\" IF SaveDialog.Execute THEN richedit1.SaveToFile SaveDialog.FileName end if g53: end sub </pre>	

26.3.2 Cursor List (Dialog Box)

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$Include <Rapidq.inc> \$Include "QCursorListBox.inc" dim i as single Declare Sub CBClick(Sender as QListBox) Create MainForm as QForm Caption = "MainForm":Top = 131:Left = 217:Width = 400 Height = 309 End Create Create CB as QCursorListBox Parent = MainForm Top = 32 Left = 80 Width = 217 Height = 177 Color = &HFFFFFF OnClick = CBClick Update End Create MainForm.ShowModal Sub CBClick(Sender as QListBox) CB.InheritOnClick MainForm.Cursor = CB.Cursor showmessage str\$(cb.cursor) MainForm.Caption = CB.CursorName End Sub </pre>	



26.3.3 Prevent double run (only one instance may be opened)

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$include <rapidq2.inc> CONST rqClassName = "TForm" CONST myFormName = "Whatever" DECLARE FUNCTION FindWindow LIB "USER32" ALIAS "FindWindowA" _ (className AS STRING, windowName AS STRING) AS INTEGER DECLARE SUB SetForegroundWindow LIB "USER32" ALIAS "SetForegroundWindow" _ (HWnd AS LONG) DECLARE SUB ShowWindow LIB "USER32" ALIAS "ShowWindow" _ (HWnd AS LONG, nCmd AS LONG) DEFINT hWnd = FindWindow(rqClassName, myFormName) IF hWnd THEN ShowMessage ["A previous instance was detected"] SetForegroundWindow(hWnd) ShowWindow(hWnd, 1) END END IF CREATE Form AS QFORM Caption = myFormName ShowModal END CREATE </pre>	<p>Try to run two of this program to see result. It will detect the FORM CAPTION NAME to be a double run.</p>

26.3.4 Alter Desktop Wallpaper

SNO	CODING	EXPLANATION
1	<pre> \$typecheck on \$INCLUDE "Rapidq.inc" \$INCLUDE "QDesktop.inc" dim Desktop as QDesktop dim file as QPENDIALOG file.Filter="Fichiers bitmap[* .bmp]*.bmp" file.InitialDir=CurDir\$ if file.Execute then Desktop.ChangeWallPaper(file.fileName) end if showmessage "done" </pre>	<p>Ensure correct size of desired wallpaper (1300x800)</p>

26.4 ELEMENT ENHANCMENT

26.4.1 QTab Control with images

SNO	CODING	EXPLANATION
1	<pre> \$TypeCheck ON \$Include "RapidQ.inc" \$Resource bmp1 As "c:\ms3\mybis\gx\glad.bmp" \$Resource bmp2 As "c:\ms3\mybis\gx\sad.bmp" Type TC_ITEM mask As Integer lpReserved1 As Long lpReserved2 As Long pszText As Long cchTextMax As Integer iImage As Integer IParam As Long End Type Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" _ (hWnd As Long, nIndex As Long, dwNewLong As Long) As Long Declare Function SendMessageAPI Lib "user32" Alias "SendMessageA" _ (hWnd As Long, Msg As Long, wParam As Long, lParam As Long) As Long Declare Function GetParent Lib "user32" Alias "GetParent" (hWnd As Long) As Long Declare Sub Form_OnClose Declare Sub MenuClick(Sender As QMenuItem) Declare Sub TabChanged </pre>	<p>Add images (BMP or ICON) to the tab control tabs.</p>

SNO	CODING	EXPLANATION
	<pre> Const GVL_STYLE = [-16] Const WS_CHILD = &H40000000:Const WS_VISIBLE = &H10000000 Const WS_CLIPSIBLINGS = &H4000000 Const TCS_MULTILINE = &H200 Const WM_SETFONT = &H30 Const TCM_SETIMAGELIST = &H1303 Const TCM_GETITEMCOUNT = &H1304 Const TCM_DELETEITEM = &H1308 Const TCM_GETCURFOCUS = &H132F Const TCM_SETCURFOCUS = &H1330 Const TCM_SETITEMW = &H133D Const TCIF_IMAGE = &H2 DefLng lRet DefLng hTabCtrl DefLng NewStyle = WS_CHILD Or WS_VISIBLE Or WS_CLIPSIBLINGS Or TCS_MULTILINE Dim TCi As TC_ITEM Dim TabLabelFont As QFont TabLabelFont.Name = "Tahoma" TabLabelFont.Size = 9 TabLabelFont.Bold = True Dim InactiveTabFont As QFont InactiveTabFont.Name = "Arial" InactiveTabFont.Color = RGB(255,255,255) InactiveTabFont.Bold = True Dim TabImageList As QImageList TabImageList.Width = 18 TabImageList.Height = 18 TabImageList.AddBMPHandle(bmp1, RGB(255,000,255)) TabImageList.AddBMPHandle(bmp2, RGB(255,000,255)) Create Form As QForm BorderStyle = bsSingle Caption = " TabControl Icon Demo" DelBorderIcons(biMinimize) Width = 320:Height = 240:Center:OnClose = Form_OnClose Create MainMenu As QMainMenu Create File As QMenuItem Caption = "&File" Create fNewTab As QMenuItem Caption = "New Tab":OnClick = MenuClick End Create Create fLine1 As QMenuItem Caption = "." End Create Create fCloseTab As QMenuItem Caption = "Close Tab":OnClick = MenuClick End Create Create fLine2 As QMenuItem Caption = "." End Create Create fExit As QMenuItem Caption = "Exit":OnClick = MenuClick End Create End Create Create TabControl As QTabControl Align = alClient:Color = RGB(202,200,196) Cursor = crHandPoint:Font = TabLabelFont Hint = "TabControl with Icons" HotTrack = True ShowHint = True TabInactiveFont = InactiveTabFont TabInactiveColor = RGB(000,000,000) AddTabs("Tab 1", "Tab 2") OnChange = TabChanged Create Edit As QEdit AutoSize = False:Align = alClient Hint = "QEdit is displayed when Tab 1 is selected" Text = "This is a QEdit in "+TabControl.Tab(0) End Create End Create </pre>	



SNO	CODING	EXPLANATION
	<pre> ShowHint = True:Visible = True End Create Create Label As QLabel AutoSize = False:Align = alClient Cursor = crArrow:Color = RGB(255,255,240) Hint = "QLabel is displayed when Tab 2 is selected" Caption = "This is a QLabel in "+TabControl.Tab(1) ShowHint = True:Visible = False End Create End Create Create StatusBar As QStatusBar SizeGrip = False:AddPanels "" Panel(0).Alignment = taCenter End Create End Create hTabCtrl = GetParent(Edit.Handle) SetWindowLong(hTabCtrl, GWL_STYLE, NewStyle) SendMessageAPI(hTabCtrl, WM_SETFONT, TabLabelFont.Handle, 1) SendMessageAPI(hTabCtrl, TCM_SETIMAGELIST, 0, TabImageList.Handle) TCI.mask = TCIF_IMAGE TCI.ilimage = 0 ' Index into the ImageList SendMessageAPI(hTabCtrl, TCM_SETITEMW, 0, TCI) TCI.mask = TCIF_IMAGE TCI.ilimage = 1 ' Index into the ImageList SendMessageAPI(hTabCtrl, TCM_SETITEMW, 1, TCI) Form.ShowModal Sub Form_OnClose Application.Terminate End Sub Sub MenuClick(Sender As QMenuItem) Select Case Sender.Handle Case fNewTab.Handle IRet = SendmessageAPI(hTabCtrl, TCM_GETITEMCOUNT, 0, 0) TabControl.AddTabs("TAB "+Str\$(IRet+1)) TCI.mask = TCIF_IMAGE TCI.ilimage = IRet Mod 2 SendMessageAPI(hTabCtrl, TCM_SETITEMW, IRet, TCI) SendMessageAPI(hTabCtrl, TCM_SETCURFOCUS, IRet, 0) Case fCloseTab.Handle IRet = SendmessageAPI(hTabCtrl, TCM_GETITEMCOUNT, 0, 0) If IRet > 2 Then IRet = SendMessageAPI(hTabCtrl, TCM_GETCURFOCUS, 0, 0) If IRet > 1 Then SendMessageAPI(hTabCtrl, TCM_DELETEITEM, IRet, 0) IRet = SendmessageAPI(hTabCtrl, TCM_GETITEMCOUNT, 0, 0) SendMessageAPI(hTabCtrl, TCM_SETCURFOCUS, IRet-1, 0) End If End If Case fExit.Handle Application.Terminate End Select End Sub Sub TabChanged IRet = SendmessageAPI(hTabCtrl, TCM_GETCURFOCUS, 0, 0) Edit.Visible = IIF(IRet = 0, True, False) Label.Visible = IIF(IRet = 1, True, False) End Sub </pre>	

26.4.2 Image StatusBar & Keyboard reading



SNO	CODING	EXPLANATION
1	<pre> \$OPTIMIZE ON \$APPTYPE GUI \$TYPECHECK ON CONST CRLF=CHR\$(13) & CHR\$(10) CONST VK_CAPITAL = &H14 CONST VK_NUMLOCK = &H90 CONST VK_INSERT = &H2D DEFSTR CURSPRV DEFBYTE MODPREV,CAPSPREV,NUMPREV,INSRTSTAT DECLARE SUB DOTIMER DECLARE SUB DOINIT DECLARE SUB INSERTSTAT(key AS byte,shift AS byte) DECLARE FUNCTION SetWindowLong LIB "User32" ALIAS _ "SetWindowLongA" (hwnd AS Long, nIndex AS Long, _ dwNewLong AS Long) AS LONG DECLARE FUNCTION SetParent LIB "user32" ALIAS _ "SetParent" (hWndChild AS Long, hWndNewParent _ AS Long) AS LONG DECLARE FUNCTION GetKeyState LIB "user32" ALIAS _ "GetKeyState" (nVirtKey AS Long) AS INTEGER CREATE TIMEX AS QTIMER INTERVAL=50 ONTIMER=DOTIMER END CREATE CREATE RFONT AS QFONT NAME="LETTER GOTHIC" SIZE=10 END CREATE CREATE SBFONT AS QFONT NAME="ARIAL" SIZE=8 BOLD=1 END CREATE CREATE FORM AS QFORM CAPTION="STATUSBAR BITMAPS DEMO" ONSHOW =DOINIT:FONT=SBFONT WIDTH = 490:HEIGHT = 520:CENTER CREATE MenuBar1 AS QMAINMENU CREATE FileMenu AS QMENUITEM Caption = "&File" END CREATE CREATE EditMenu AS QMENUITEM Caption = "&Edit" END CREATE CREATE ToolsMenu AS QMENUITEM Caption = "&Tools" END CREATE CREATE HelpMenu AS QMENUITEM Caption = "&Help" END CREATE END CREATE CREATE REDIT AS QRICHEDIT LEFT = 0:TOP = 89:WIDTH = 532:HEIGHT = 382:ALIGN = 5 FONT=RFONT:ONKEYDOWN=INSERTSTAT:HIDSELECTION = 0 PLAINTEXT = 1:SCROLLBARS = 3:WANTTABS = 1:TABORDER = 0 END CREATE CREATE STATUSBAR AS QSTATUSBAR ADDPANELS "" , "" , "" :PANEL(0).WIDTH = 185 PANEL(1).WIDTH = 180 PANEL(2).CAPTION="C:\RapidQ\Examples\zipview.rqb" END CREATE CREATE STPANEL AS QPANEL HEIGHT=STATUSBAR.HEIGHT-5:BEVELOUTER=0:TOP=4 CREATE IMAGE1 AS QIMAGE TRANSPARENT=1:LEFT=0:BMP="c:\ms3\mybis\gx\INSERTONSG.BMP" END CREATE CREATE IMAGE2 AS QIMAGE TRANSPARENT=1:LEFT=44:BMP="c:\ms3\mybis\gx\CAPSONSG.BMP" END CREATE </pre>	<p>Add images to the status bar and READ THE KEYBOARD STATUS.</p>



SNO	CODING	EXPLANATION
	<pre> CREATE IMAGE3 AS GIMAGE TRANSPARENT=1:LEFT=88:BMP="c:\ms3\mybis\gx\NUMONSG.BMP" END CREATE CREATE IMAGE4 AS GIMAGE TRANSPARENT=1:LEFT=132:BMP="c:\ms3\mybis\gx\SAVEDSQ.BMP" END CREATE END CREATE END CREATE SUB DOINIT SetParent Stpanel.Handle , StatusBar.Handle STPANEL.WIDTH=StatusBar.PANEL{1}.WIDTH - 4 END SUB ' SUB INSERTSTAT(KEY,SHIFT) DEFBYTE KEYST IF KEY=45 AND SHIFT=0 THEN INSRTSTAT=1-INSRTSTAT IF INSRTSTAT=0 THEN IMAGE1.BMP="c:\ms3\mybis\gx\INSERTONSG.BMP" ELSE IMAGE1.BMP="c:\ms3\mybis\gx\INSERTOFFSQ.BMP" END IF END IF END SUB SUB DOTIMER DEFSTR CAPSTR,CURSTR CURSTR=STR\$(REDIT.WHEREY+1) & ". " & _ STR\$(REDIT.WHEREX+1) _ & " / " & STR\$(REDIT.LINECOUNT) & _ " Lines, Size=" & STR\$(len(REDIT.text)) IF CURSPRV<CURSTR THEN CURSPRV=CURSTR STPANEL.FONT=SBFONT StatusBar.PANEL{0}.CAPTION=CURSTR StatusBar.PANEL{0}.WIDTH=FORM.TEXTWIDTH(CURSTR)+6 STPANEL.LEFT=StatusBar.PANEL{0}.WIDTH + 3 END IF IF MODPREV<>REDIT.MODIFIED THEN MODPREV=REDIT.MODIFIED IF REDIT.MODIFIED THEN IMAGE4.BMP="c:\ms3\mybis\gx\MODIFYSQ.BMP" ELSE IMAGE4.BMP="c:\ms3\mybis\gx\SAVEDSQ.BMP" END IF END IF IF CAPSPREV<>GetKeyState[VK_CAPITAL] OR _ NUMPREV<>GetKeyState[VK_NUMLOCK] THEN CAPSPREV=GetKeyState[VK_CAPITAL] NUMPREV=GetKeyState[VK_NUMLOCK] IF CAPSPREV THEN IMAGE2.BMP="c:\ms3\mybis\gx\CAPSONSG.BMP" ELSE IMAGE2.BMP="c:\ms3\mybis\gx\CAPSOFFSQ.BMP" END IF IF NUMPREV THEN IMAGE3.BMP="c:\ms3\mybis\gx\NUMONSG.BMP" ELSE IMAGE3.BMP="c:\ms3\mybis\gx\NUMOFFSQ.BMP" END IF END IF END SUB SetWindowLong FORM.HANDLE, -8, 0 SetWindowLong application.HANDLE, -8, FORM.HANDLE FORM.SHOWMODAL </pre>	

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> const REO_NULL = 0 Const REO_READWRITEMASK = &H3F const REO_DONTNEEDPALETTE = &H20 Const REO_BLANK = &H10 const REO_DYNAMICSIZE = &H08 const REO_INVERTEDSELECT = &H04 const REO_BELOWBASELINE = &H02 Const REO_RESIZABLE = &H01 CONST EM_GETOLEINTERFACE = &H400 + 60 Declare Function InsertBitmapFromFile Lib "REImage.dll" Alias "InsertBitmapFromFile" (rEdit as long, FileName as string, dwFlags as long) as integer Declare Function InsertBitmapFromData Lib "REImage.dll" Alias "InsertBitmapFromData" (rEdit as long, hBitmap as long, dwFlags as long) as integer Declare sub Form_Load CREATE Form AS QFORM Caption = "Form1":Width = 320:Height = 240:Center CREATE RichEdit1 AS QRICHEDIT Left = 0:Top = 0:Width = 312:Height = 213:Align = 5 plainText = false:AddStrings "RichEdit1" END CREATE onShow = Form_Load END CREATE Form.ShowModal sub Form_Load dim s as string s = "c:\ms3\mybis\gx\Spiral.bmp" InsertBitmapFromFile(RichEdit1.Handle, s , REO_RESIZABLE) End Sub </pre>	<p>You need to include the following files with your application:</p> <ul style="list-style-type: none"> • Reimage.cpp • Reimage.dll • Reimage.exw • Reimage.hpp

26.4.4 QRichEdit spell check and font colouring

SNO	CODING	EXPLANATION
1	<pre> \$APPTYPE GUI \$TYPECHECK ON \$INCLUDE "RAPIDQ.INC" TYPE QREdit EXTENDS QRichEdit RichFont AS QFont Syntaxes(100) AS STRING MaxSyntax AS INTEGER HiLightColor AS INTEGER SUB HiLight DIM TempStart AS INTEGER:DIM I AS INTEGER, N AS INTEGER WITH QREdit TempStart = .SelStart .SelStart = 0 .SelLength = LEN(.Text) .SelAttributes = .RichFont FOR I = 1 TO .MaxSyntax N = INSTR(UCASE\$(.Text), UCASE\$(.Syntaxes[I]))-1 WHILE N >= 0 .SelStart = N .RichFont.AddStyles(fsBold) .RichFont.Color = .HiLightColor .SelLength = LEN(.Syntaxes[I]) .SelAttributes = .RichFont N = INSTR(N+.SelLength, UCASE\$(.Text), UCASE\$(QREdit.Syntaxes[I]))-1 WEND .SelLength = 0 .RichFont.DelStyles(fsBold) .RichFont.Color = 0 .Font = .RichFont .SelStart = TempStart NEXT I END WITH END SUB </pre>	<p>This will confirm the word as identified and highlight them.</p>



SNO	CODING	EXPLANATION
	<pre> SUBI AddSyntaxes(...) DIM I AS INTEGER WITH QREdit FOR I = 1 TO ParamStrCount .Syntaxes(I+.MaxSyntax) = ParamStr\$(I) NEXT .MaxSyntax = .MaxSyntax + ParamStrCount END WITH END SUBI EVENT OnKeyUp (Key AS WORD, Shift AS INTEGER) DIM I AS INTEGER, EndStr AS INTEGER, StartStr AS INTEGER DIM TempStart AS INTEGER, N AS INTEGER DIM Token AS STRING DIM T1 AS INTEGER, T2 AS INTEGER T1 = QREdit.SelStart:T2 = QREdit.SelLength IF Key < 46 AND Key >= 8 THEN EXIT EVENT END IF WITH QREdit FOR I = .SelStart TO LEN(.Text) IF MID\$(.Text, I, 1) = " " OR MID\$(.Text, I, 1) = CHR\$(13) OR MID\$(.Text, I, 1) = CHR\$(10) THEN EXIT FOR END IF NEXT I EndStr = I FOR I = .SelStart TO 1 STEP -1 IF MID\$(.Text, I, 1) = " " OR MID\$(.Text, I, 1) = CHR\$(10) OR MID\$(.Text, I, 1) = CHR\$(13) THEN EXIT FOR END IF NEXT I StartStr = I+1 Token = RTRIM\$(LTRIM\$(MID\$(.Text, StartStr, EndStr - StartStr))) TempStart = .SelStart .SelStart = StartStr-1 .SelLength = LEN(Token) .SelAttributes = .RichFont FOR I = 1 TO .MaxSyntax IF UCASE\$(Token) = UCASE\$(.Syntaxes(I)) THEN .SelStart = StartStr-1 .RichFont.AddStyles(fsBold) .RichFont.Color = .HiLightColor .SelLength = LEN(.Syntaxes(I)) .SelAttributes = .RichFont END IF NEXT I .SelLength = 0 .SelStart = TempStart .RichFont.DelStyles(fsBold) .RichFont.Color = 0 .Font = .RichFont END WITH QREdit.SelStart = T1 QREdit.SelLength = T2 END EVENT CONSTRUCTOR PlainText = True RichFont.Name = "Courier" MaxSyntax = 0 HiLightColor = &HAA0000 Font = QREdit.RichFont END CONSTRUCTOR END TYPE CREATE Form AS QForm CREATE RichEdit AS QREdit Align = alClient WordWrap = False Clear AddSyntaxes("PRINT", "CLS", "LOCATE", "END", "COLOR", "IF", "THEN", "ELSE", "RAP", "MILDA") </pre>	



SNO	CODING	EXPLANATION
	<pre>AddStrings("Syntax Highlighting", "", _ "10 CLS", _ "20 PRINT "+CHR\$(34)+"Hello World!"+CHR\$(34), _ "30 LOCATE 10,10", _ "40 PRINT 1+2+3", _ "50 IF RAPIDQ = GREAT THEN", _ "60 PRINT "+CHR\$(34)+"Oh yeah!"+CHR\$(34), _ "70 ELSE", _ "80 END", _ "90 END IF") HiLight END CREATE Caption = "Syntax highlighting" Center ShowModal END CREATE</pre>	

26.45 View all images (image dialog box)

SNO	CODING	EXPLANATION
1	<pre>\$typecheck on \$include "QImageDialog.inc" dim a as QImageDialog a.filter="format[*.*bmp;*.ico;*.ani;*.cur*.gif;*.jpg;*.dib;*.rle;*.tga;*.pcx]**.bmp;*.ico;*.ani;*.cur*.gif;*.jpg;*.dib;*.rle;*.tga;*.pcxFichiers bitmap[*.*bmp]**.bmpFichiers gif[*.*gif]**.gifFichiers jpg[*.*jpg]**.jpgFichiers ico[*.*ico]**.icoFichiers curseur[*.*cur;*.ani]**.cur;*.ani" a.caption="IMAGE DIALOG" a.DlgCaption(1)="Files:" a.DlgCaption(2)="Type Format:" a.DlgCaption(3)="Open" a.DlgCaption(4)="Cancel!" a.errorLanguage="english" a.MultiSelect=true a.execute end</pre>	

26.46 Tree Viewing Module

SNO	CODING	EXPLANATION
1	<pre>\$TYPECHECK ON SUB TreeViewChange (Node AS INTEGER, AllowChange AS INTEGER, Sender AS QTREEVIEW) IF Node = 8 THEN AllowChange = 0 END SUB SUB TreeViewMouseMove (X AS INTEGER, Y AS INTEGER, Shift AS INTEGER, Sender AS QTREEVIEW) dim i as single I = Sender.GetItemAt(X,Y) IF I >= 0 THEN Sender.ItemIndex = I END SUB CREATE Form AS QFORM Center CREATE TreeView AS QTREEVIEW Align = 5: AddItems "1","2","3":AddChildItems 0, "Sub 1", "Sub 2", "Sub 3" AddChildItems 4, "Sub 1", "Sub 2", "Sub 3":FullExpand OnChangeing = TreeViewChange:OnMouseMove = TreeViewMouseMove END CREATE END CREATE dim i as single FOR I = 0 TO TreeView.ItemCount-1 TreeView.Item(I).Text = STR\$(I) NEXT</pre>	

