

MODULE 16: SOUND & MUSIC

On completion of this module you will be able to use FX, music and sound with your program. Both recording and playing thereof will be made possible.

MODULE 16.1: SOUND & MUSIC

Subject Outcome 1: Introduction

Subject Outcome 2: WAVE sounds/music Files

Subject Outcome 3: MIDI Files

Subject Outcome 4: Display Sound Control Panel

Subject Outcome 5: Recording sound (via Mic/Line In Jack)

Subject Outcome 6: QWAVE (incl high quality recording & MP3 playing)

Subject Outcome 7: MP3

Subject Outcome 8: Voice

Subject Outcome 9: CD Player

Subject Outcome 10: MP3 Tags

16.1 INTRODUCTION

Sound and Music are two aspects that no program can operate without. It is not so much that the program will not work, it just give life to your program. What complicate sound are the different formats (MP3, WAV, OGG, MIDI, WAV, etc.)



Some formats you are not allowed to use at all without registering with the 3rd party owner. The sound methods that we will address may be used and is fully compatible with FREEQ. Take note however that the actual sound/music that you play has its own rights and you should address to their copyrights ©. Only 1 WAV file can play at a time.

16.2 WAVE SOUNDS/MUSIC

16.2.1 Direct Method

This will play a WAV or file. A WAV file is a type, the most commonly used sound file as it is compatible with all sounds programs, of sound file. It may either be a sound (FX) or music file. The file size is however a large size, much larger than MP3. **The layout:**

Playwav "...", 1 or 3

"..." represents file (include the path).

1 represents the influence of playing:

- 0 wait for sound to finish playing before executing next command.
- 1 play in the background (program continue while playing).

or 3 represents follow up:

- Or 0 don't repeat.
- Or 3 repeat once.



SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> declare sub playwavnow CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480: Center create but1 as qbutton left=10:top=10:caption="Play":onclick=playwavnow end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal sub playwavnow playwav "c:\ms3\mybis\gx\ e28logo.wav",1 or 0 end sub </pre>	<p>Sub program that will run the file.</p> <p>The button to be pressed to play the WAV file.</p> <p>Play the WAV file, in background and don't repeat.</p>

16.2.2 Resource Method

This will play a WAV or file from a resource file that has been embedded into your program. Be careful of this method as the EXE file size will be increased drastically.

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$resource start1 as "c:\ms3\mybis\gx\ e28logo.wav" declare sub playwavnow CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480: Center create but1 as qbutton left=10:top=10:caption="Play" onclick=playwavnow end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal sub playwavnow playwav start1,1 or 0 end sub </pre>	<p>Embed into your program.</p> <p>Play the resource file call START1 (take note that pending on the size of the size of the file, it could maybe a second or two to start playing.</p>

16.3 MIDI Files

MIDI files are created by studio recording equipment (electronic keyboards, etc.) with software written specifically for that equipment. The output file is MIDI files. You can only play 1 MIDI sequence at a time.



SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <qmidi.inc> dim midme as qmidi declare sub playnow CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480: Center create but1 as qbutton left=10:top=10:caption="Play" onclick=playnow end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) midme.open("c:\ms3\mybis\gx\paper01.mid") Form.ShowModal sub playnow midme.play end sub </pre>	<p>Establishing the MIDI and opening it.</p> <p>Include the capability. Establish the Midi Element</p> <p>Load the file - no playing yet.</p>
2	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <qmidi.inc> dim midme as qmidi declare sub playnow declare sub stopnow CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480: Center create but1 as qbutton left=10:top=10:caption="Play" onclick=playnow end create create pauseme as qbutton left=200:top=10:caption="Stop":onclick=stopnow end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) midme.open("c:\ms3\mybis\gx\paper01.mid") Form.ShowModal sub playnow if but1.caption="Play" then midme.play:but1.caption="Pause" goto endplaynow end if if but1.caption="Pause" then but1.caption="Play":midme.pause end if endplaynow: end sub sub stopnow midme.stop:but1.caption="Play" end sub </pre>	<p>Pause and Stop methods.</p>



SNO	CODING	EXPLANATION
3	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <qmidi.inc> dim midme as qmidi declare sub confirmstatus declare sub playnow declare sub stopnow CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480: Center create but1 as qbutton left=10:top=10:caption="Play" onclick=playnow end create create pauseme as qbutton left=200:top=10:caption="Stop":onclick=stopnow end create create remark as qlabel left=20:top=380 end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) midme.open("c:\ms3\mybis\gx\paper01.mid") confirmstatus Form.ShowModal sub playnow if but1.caption="Play" then midme.play:but1.caption="Pause" confirmstatus goto endplaynow end if if but1.caption="Pause" then but1.caption="Play":midme.pause confirmstatus end if endplaynow: end sub sub stopnow midme.stop:but1.caption="Play" confirmstatus end sub sub confirmstatus if midme.fileopen=0 then form.caption="No file" if midme.fileopen=1 then form.caption="File Opened" if midme.state=0 then remark.caption="Closed" if midme.state=1 then remark.caption="Playing ..." if midme.state=2 then remark.caption="Paused" if midme.state=3 then remark.caption="Stopped" end sub </pre>	<p>Determine status of MIDI element. You have FILEOPEN to determine if a file has been loaded or not and then the STATE:</p> <ul style="list-style-type: none"> • 0: Closed • 1: Play • 2: Pause • 3: Stopped <p>Whenever the status has been altered, then the sub program is called to display the current status.</p>
4	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <qmidi.inc> dim midme as qmidi declare sub confirmstatus declare sub playnow declare sub stopnow CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480: Center </pre>	<p>Lenght. The length of the actual MIDI sequence is determined and stored in the LENGTH numerical method (variable). The value is indicated in values of a 1000. One second therefore = 1000 frames.</p>



SNO	CODING	EXPLANATION
	<pre> create but1 as qbutton left=10:top=10:caption="Play" onclick=playnow end create create pauseme as qbutton left=200:top=10:caption="Stop":onclick=stopnow end create create remark as qlabel left=20:top=380 end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) midme.open("c:\ms3\mybis\gx\paper01.mid") confirmstatus Form.ShowDialog sub playnow if but1.caption="Play" then midme.play:but1.caption="Pause" confirmstatus goto endplaynow end if if but1.caption="Pause" then but1.caption="Play":midme.pause confirmstatus end if endplaynow: end sub sub stopnow midme.stop:but1.caption="Play" confirmstatus end sub sub confirmstatus if midme.fileopen=0 then form.caption="No file" if midme.fileopen=1 then form.caption="File Opened: "+str\$(fix(midme.lenght/1000)) end if if midme.state=0 then remark.caption="Closed" if midme.state=1 then remark.caption="Playing ..." if midme.state=2 then remark.caption="Paused" if midme.state=3 then remark.caption="Stopped" end sub </pre>	<p>The length is divided by 1000 to determine the total seconds. Note the spelling of LENGHT.</p>
5	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <qmidi.inc> dim midme as qmidi dim place as single declare sub confirmstatus declare sub playnow declare sub stopnow declare sub updatetrack(xposbar as long) CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480: Center create but1 as qbutton left=10:top=10:caption="Play" onclick=playnow end create create pauseme as qbutton </pre>	<p>ONCHANGE=SS\$?TIMER.INTERVAL=n%</p> <p>This will activate an event monitor on whenever the position (playing position) is altered. We will use a trackbar to display the position of the song being played. Note that we need to determine the % position as the trackbar is 0-100.</p> <p>The formula used: Place=([U*100]/midme.lenght)</p> <p>Timer.Interval will determine how long before confirming the current playing position. 1 Sec = 1000 beats.</p>

SNO	CODING	EXPLANATION
	<pre> left=200:top=10:caption="Stop":onclick=stopnow end create create remark as qlabel left=20:top=380 end create create barpositn as qtrackbar left=20:top=150:width=450:max=100 end create END CREATE SetWindowLong[Form.Handle, -8, 0] SetWindowLong[Application.Handle, -8, Form.Handle] midme.open("c:\ms3\mybis\gx\paper01.mid") confirmstatus midme.onchange=updatetrack midme.timer.interval=250 Form.ShowModal sub playnow if but1.caption="Play" then midme.play:but1.caption="Pause" confirmstatus goto endplaynow end if if but1.caption="Pause" then but1.caption="Play":midme.pause confirmstatus end if endplaynow: end sub sub stopnow midme.stop:but1.caption="Play" confirmstatus end sub sub confirmstatus if midme.fileopen=0 then form.caption="No file" if midme.fileopen=1 then form.caption="File Opened: "+str\$(fix(midme.lenght/ 1000)) end if if midme.state=0 then remark.caption="Closed" if midme.state=1 then remark.caption="Playing ..." if midme.state=2 then remark.caption="Paused" if midme.state=3 then remark.caption="Stopped" barpositn.position=0 end if end sub sub updatetrack(xposbar as long) place=((xposbar* 100)/midme.lenght) barpositn.position=place remark.caption="position: "+str\$(xposbar/1000) end sub </pre>	<p>Activate event handler. Update every ¼ of a second.</p>
6	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <qmidi.inc> dim midme as qmidi dim place as single declare sub newpos declare sub confirmstatus </pre>	<p>Manual altering position of where the song is playing. This is done by altering the position of the trackbar and then re-allocating the position of the play frame. Song must be paused/stopped when you move the trackbar – press PLAY to continue</p>



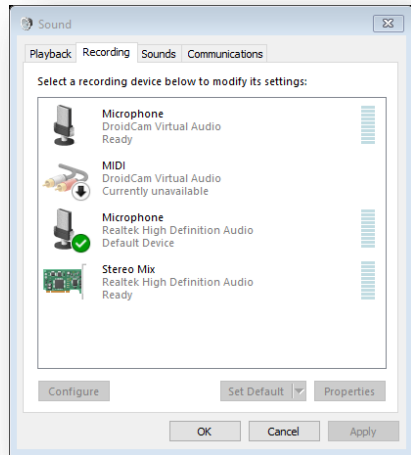
SNO	CODING	EXPLANATION
	<pre> declare sub playnow declare sub stopnow declare sub updatetrack(xposbar as long) CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480: Center create but1 as qbutton left=10:top=10:caption="Play":onclick=playnow end create create pauseme as qbutton left=200:top=10:caption="Stop":onclick=stopnow end create create remark as qlabel left=20:top=380 end create create barpositn as qtrackbar left=20:top=150:width=450:max=100:onChange=newpos end create END CREATE SetWindowLong[Form.Handle, -8, 0] SetWindowLong[Application.Handle, -8, Form.Handle] midme.open["c:\ms3\mybis\gx\paper01.mid"] confirmstatus midme.onChange=updatetrack:midme.timer.interval=250 Form.ShowModal sub playnow if but1.caption="Play" then midme.play:but1.caption="Pause":confirmstatus goto endplaynow end if if but1.caption="Pause" then but1.caption="Play":midme.pause:confirmstatus end if endplaynow: end sub sub stopnow midme.stop:but1.caption="Play" confirmstatus end sub sub confirmstatus if midme.fileopen=0 then form.caption="No file" if midme.fileopen=1 then form.caption="File Opened: "+str\$(fix(midme.lenght/ 1000)) end if if midme.state=0 then remark.caption="Closed" if midme.state=1 then remark.caption="Playing ..." if midme.state=2 then remark.caption="Paused" if midme.state=3 then remark.caption="Stopped":barpositn.position=0 end if end sub sub updatetrack(xposbar as long) place=[(xposbar* 100)/midme.lenght] barpositn.position=place remark.caption="position: "+str\$(xposbar/ 1000) end sub sub newpos midme.currentframe=(barpositn.position/ 100)*midme.lenght end sub </pre>	<p>Call the sub program to reallocate the playing position.</p> <p>Formula to reallocate the playing position.</p>



SNO	CODING	EXPLANATION
7	<pre>... Sub Closenow Midme.stop Midme.close Barpositn.position=0 End sub ...</pre>	<p>Closing a MIDI sequence. If you wish to load another MIDI sequence, then you must stop the current sequence, then close it.</p>
8	<pre>\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <qmidi.inc> dim midme as qmidi CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480:Center END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Midme.open("c:\ms3\mybis\gx\paper01.mid") Midme.Play Playwav("c:\ms3\mybis\gx\right.wav",1 or 0) Form.ShowModal</pre>	<p>Playing a WAVE file and MIDI file at the same time. This is possible, as long as the WAVE file is played in the background and not pausing the program until finished playing the WAVE sequence.</p>

16.4 Sound Control Panel

The **SOUNDS Control Panel** is the master control system for the **entire sound settings** of the **computer**. You will need to call the **QSYSTEM** component.

SNO	CODING	EXPLANATION
1	<pre>\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim syst as qsystem declare sub callsound CREATE Form AS QFORM Caption = "SOUND" Width = 640:Height = 480:Center create menu as qmainmenu create menua as qmenuitem caption="Sound":onclick=callsound end create end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal sub callsound syst.showcontrolpanel("sounds",1) end sub</pre>	<p>Establish the QSYSTEM component/element.</p>  <p>Call and display the control panel for sounds.</p>

16.5 Recording Sound (via Mic/Line In Jack/Stereo Mix)

See QWAVE for better quality recording. During this lesson you will be taught how to record sound using a MIC/Line Input (radio or any audio system coupled to the computer – AUX) or any other means of audio input to the computer. Once recording has completed, it will be saved as a WAV file. Please take note that you need to ensure that the RECORDING settings are correct (see 16.4). You may record either from the MIC or LINE INPUT or Stereo Mix (internal sound being played). We will now code the program to record and then play back what was recorded (both user input buttons to be pressed).

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim syst as qsystem declare sub callsound declare sub startrecord declare sub playback DECLARE FUNCTION mciSendString LIB "WINMM.DLL" ALIAS "mciSendStringA" _ (StrCommand AS STRING, ReturnString AS LONG, _ ReturnLength AS LONG, hwndCallBack AS LONG) AS INTEGER CREATE Form AS QFORMex Caption = "SOUND" Width = 640:Height = 480:Center create menu as qmainmenu create menua as qmenuitem caption="Sound":onclick=callsound end create end create create record as qbutton left=50:top=50:caption="RECORD":onclick=startrecord end create create playbackbtn as qbutton left=150:top=50:caption="PLAY":enabled=0 onclick=playback end create END CREATE SetWindowLong[Form.Handle, -8, 0] SetWindowLong[Application.Handle, -8, Form.Handle] Form.ShowModal sub callsound syst.showcontrolpanel("sounds",1) end sub sub startrecord if record.caption="RECORD" then mciSendString "open new Type waveaudio Alias capture", 0, 0, 0 mciSendString "set capture bitspersample 8", 0, 0, 0 mciSendString "set capture samplespersec 11025", 0, 0, 0 mciSendString "set capture channels 1", 0, 0, 0 mciSendString "record capture", 0, 0, 0 record.caption="....." goto endstartrecord end if </pre>	<p>You may record from MIC, Line In Jack (AUX) or Sound Mixer (internal sound) – set at sound settings.</p> <p>Sub program to record. Sub program to play back sound. Include the DLL file that handles recording.</p> <p>The record button. When we press it, the caption will alter as "....." to indicate it is busy recording. Press again to STOP recording – when you stop, it will auto save the file.</p> <p>The Play button.</p> <p>The recording sub program. If button is pressed to RECORD, then start recording in 8 bit mode at samples per seconds 11025. Channel is 1 (if 2 then stereo mode is on). Start capturing. Alter button caption to indicate busy with recording.</p>



SNO	CODING	EXPLANATION
	<pre> if record.caption="....." then mciSendString "stop capture", 0, 0, 0 mciSendString "save capture c:\ms3\~common\recw5.wav", 0, 0, 0 mciSendString "close capture", 0, 0, 0 record.caption="RECORD" playbackbtn.enabled=1 end if endstartrecord: end sub sub playback playwav ("c:\ms3\~common\recw5.wav",1) end sub </pre>	<p>If the busy recording button is active, then STOP recording. Save as the file indicated. Close the recording. Alter button to wait for new RECORDING.</p> <p>Play back the recorded file.</p>

16.6 QWAVE (incl high quality recording & MP3 Playing)

QWAVE is a media component used to play or record wave files. You need to include the QWAVE.INC file to activate this element. QWAVE must be created outside the Source Creation Area.

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON #include <rapidq2.inc> #include <qwave.inc> dim wave as qwave create testme as qform caption="WAVER":center:width=500:height=450 end create SetWindowLong(testme.Handle, -8, 0) SetWindowLong(Application.Handle, -8, testme.Handle) testme.showmodal </pre>	<p>Activation and Assigning</p> <p>Include QWAVE element. Activate QWAVE</p>
2	<pre> \$TYPECHECK ON #include <rapidq2.inc> #include <qwave.inc> dim wave as qwave declare sub playme declare sub pauseme declare sub stopme wave.open("c:\ms3\mybis\gx\ e28logo.wav") create testme as qform caption="WAVER":center:width=500:height=450 create playbtn as qbutton left=20:top=30:caption="Play":onclick=playme end create create pausebtn as qbutton left=120:top=30:caption="Pause":enabled=0 onclick=pauseme end create create stopbtn as qbutton left=220:top=30:caption="Stop":enabled=0 onclick=stopme end create end create SetWindowLong(testme.Handle, -8, 0) SetWindowLong(Application.Handle, -8, testme.Handle) testme.showmodal </pre>	<p>Pause & Stop Modes</p> <p>Open the WAV file.</p> <p>Button to pause song.</p> <p>Button to stop song.</p>



SNO	CODING	EXPLANATION
	<pre> sub playme wave.play playbtn.enabled=0:pausebtn.enabled=1:stopbtn.enabled=1 end sub sub pauseme wave.pause playbtn.enabled=1:pausebtn.enabled=0 end sub sub stopme wave.stop playbtn.enabled=1:pausebtn.enabled=0:stopbtn.enabled=0 end sub </pre>	<p>Enable the other buttons.</p> <p>Pause the song.</p> <p>Stop the song (restart).</p>
3	<pre> \$TYPECHECK ON \$include <rapidq2.inc> \$include <qwave.inc> dim wave as qwave declare sub playme declare sub pauseme declare sub stopme declare sub alterpos(position as long) wave.open("c:\ms3\mybis\gx\ e28logo.wav") wave.onchange=alterpos create testme as qform caption="WAVER":center:width=500:height=450 create remark as qlabel left=20:top=250:caption="(pres the pause or play button)" end create create playbtn as qbutton left=20:top=30:caption="Play":onclick=playme end create create pausebtn as qbutton left=120:top=30:caption="Pause":enabled=0 onclick=pauseme end create create stopbtn as qbutton left=220:top=30:caption="Stop":enabled=0 onclick=stopme end create end create SetWindowLong(testme.Handle, -8, 0) SetWindowLong(Application.Handle, -8, testme.Handle) testme.showmodal sub playme wave.play:playbtn.enabled=0:pausebtn.enabled=1:stopbtn.enabled=1 end sub sub pauseme wave.pause:playbtn.enabled=1:pausebtn.enabled=0 end sub sub stopme wave.stop:playbtn.enabled=1:pausebtn.enabled=0:stopbtn.enabled=0 end sub sub alterpos(position as long) remark.caption="Playing: "+str\$(position) end sub </pre>	<p>OnChange (event handler as song sequence changes – position) – one second = 1000</p> <p>Create sub program</p> <p>Activate event handler (timer)</p> <p>Remember the result is in 1000 per second. Take note that you could have also used WAVE.CURRENTPOS which is a numerical variable; instead of STR\$(POSITION); STR\$(WAVE.CURRENTPOS)</p>



SNO	CODING	EXPLANATION
4	<pre> \$TYPECHECK ON \$include <rapidq2.inc> \$include <qwave.inc> dim wave as qwave declare sub playme declare sub pauseme declare sub stopme declare sub alterpos(position as long) wave.open("c:\ms3\mybis\gx\ e28logo.wav") wave.onchange=alterpos create testme as qform caption="WAVER":center:width=500:height=450 create remark as qlabel left=20:top=250:caption="(press the pause or play button)" end create create playbtn as qbutton left=20:top=30:caption="Play":onclick=playme end create create pausebtn as qbutton left=120:top=30:caption="Pause":enabled=0 onclick=pauseme end create create stopbtn as qbutton left=220:top=30:caption="Stop":enabled=0 onclick=stopme end create create gauge as qgauge left=10:top=2:height=12:width=350 end create end create SetWindowLong(testme.Handle, -8, 0) SetWindowLong(Application.Handle, -8, testme.Handle) gauge.position=0:gauge.max=wave.lenght testme.showmodal sub playme wave.play:playbtn.enabled=0:pausebtn.enabled=1:stopbtn.enabled=1 end sub sub pauseme wave.pause:playbtn.enabled=1:pausebtn.enabled=0 end sub sub stopme wave.stop:playbtn.enabled=1:pausebtn.enabled=0:stopbtn.enabled=0 end sub sub alterpos(position as long) remark.caption="Playing: "+str\$(fix(position/1000))+"/"+str\$(fix(wave.lenght/1000)) gauge.position=position end sub </pre>	<p>Lenght (note the spelling)</p> <p>This will determine the length of the sequence.</p> <p>The / 1000 is to convert the position as seconds.</p>
5	<pre> \$TYPECHECK ON \$include <rapidq2.inc> \$include <qwave.inc> dim wave as qwave declare sub playme declare sub pauseme declare sub stopme declare sub alterpos(position as long) wave.open("c:\ms3\mybis\gx\ e28logo.wav") wave.onchange=alterpos </pre>	<p>State and Information</p> <p>State:</p> <ul style="list-style-type: none"> • Play (1) • Paused (2) • Stopped (3) • Record (4)



SNO	CODING	EXPLANATION
	<pre> create testme as qform caption="WAVER":center:width=500:height=450 create remark as qlabel left=20:top=250:caption="(press the pause or play button)" end create create remark2 as qlabel left=20:top=280:caption="Mode" end create create remark3 as qlabel left=20:top=310:caption="(info)" end create create playbtn as qbutton left=20:top=30:caption="Play":onclick=playme end create create pausebtn as qbutton left=120:top=30:caption="Pause":enabled=0 onclick=pauseme end create create stopbtn as qbutton left=220:top=30:caption="Stop":enabled=0 onclick=stopme end create create gauge as qgauge left=10:top=2:height=12:width=350 end create end create SetWindowLong(testme.Handle, -8, 0) SetWindowLong(Application.Handle, -8, testme.Handle) gauge.position=0:gauge.max=wave.lenght testme.showmodal sub playme wave.play:playbtn.enabled=0:pausebtn.enabled=1:stopbtn.enabled=1 if wave.state=1 then remark2.caption="Playing" remark3.caption="Bits: "+str\$(wave.bits)+chr\$(13) remark3.caption=remark3.caption+"Freq: "+str\$(wave.frequency)+chr\$(13) if wave.mode=1 then remark3.caption=remark3.caption+"Mono Mode" end if if wave.mode=2 then remark3.caption=remark3.caption+"Stereo Mode" end if end if end sub sub pauseme wave.pause:playbtn.enabled=1:pausebtn.enabled=0 if wave.state=2 then remark2.caption="Paused" end sub sub stopme wave.stop:playbtn.enabled=1:pausebtn.enabled=0:stopbtn.enabled=0 if wave.state=3 then remark2.caption="Stopped" end sub sub alterpos(position as long) remark.caption="Playing: "+str\$(fix(position/1000))+"/"+str\$(fix(wave.lenght/1000)) if wave.state=4 then remark2.caption="Recording" gauge.position=position end sub </pre>	<p>Information:</p> <ul style="list-style-type: none"> • Bits • Frequency • Mode <p>State = Playing Info: Bits Info: Frequency</p> <p>Mono channel if = 1</p> <p>Stereo channel if = 2</p>



SNO	CODING	EXPLANATION
6	<pre> \$TYPECHECK ON \$include <rapidq2.inc> \$include <qwave.inc> dim wave as qwave declare sub playme declare sub pauseme declare sub stopme declare sub alterpos(position as long) wave.devicetype="mpegvideo" wave.open("c:\ms3\mybis\gx\runwaters.mp3") wave.onChange=alterpos create testme as qform caption="WAVER":center:width=500:height=450 create remark as qlabel left=20:top=250:caption="(press the pause or play button)" end create create remark2 as qlabel left=20:top=280:caption="Mode" end create create remark3 as qlabel left=20:top=310:caption="(info)" end create create playbtn as qbutton left=20:top=30:caption="Play":onclick=playme end create create pausebtn as qbutton left=120:top=30:caption="Pause":enabled=0 onclick=pauseme end create create stopbtn as qbutton left=220:top=30:caption="Stop":enabled=0 onclick=stopme end create create gauge as qgauge left=10:top=2:height=12:width=350 end create end create SetWindowLong(testme.Handle, -8, 0) SetWindowLong[Application.Handle, -8, testme.Handle] gauge.position=0:gauge.max=wave.lenght testme.showmodal sub playme wave.play:playbtn.enabled=0:pausebtn.enabled=1:stopbtn.enabled=1 if wave.state=1 then remark2.caption="Playing" remark3.caption="Bits: "+str\$(wave.bits)+chr\$(13) remark3.caption=remark3.caption+"Freq: "+str\$(wave.frequency)+chr\$(13) if wave.mode=1 then remark3.caption=remark3.caption+"Mono Mode" end if if wave.mode=2 then remark3.caption=remark3.caption+"Stereo Mode" end if end sub sub pauseme wave.pause:playbtn.enabled=1:pausebtn.enabled=0 if wave.state=2 then remark2.caption="Paused" end sub </pre>	<p>Playing MP3 sounds/ music.</p> <p>Must be coded before the OPEN command.</p>



SNO	CODING	EXPLANATION
	<pre> sub stopme wave.stop;playbtn.enabled=1;pausebtn.enabled=0;stopbtn.enabled=0 if wave.state=3 then remark2.caption="Stopped" end sub sub alterpos(position as long) remark.caption="Playing: "+str\$(fix(position/1000))+"/"+str\$(fix(wave.lenght/1000)) if wave.state=4 then remark2.caption="Recording" gauge.position=position end sub </pre>	
7	<pre> ... If wave.fileopen=0 Showmessage "no file loaded" End if If wave.fileopen=1 then Wave.play End if </pre>	<p>FileOpen=0 FileOpen=1 It is very important that you confirm if a file is loaded before playing it, else it will crash your program.</p>
8	<pre> \$TYPECHECK ON \$include <rapidq2.inc> \$include <qwave.inc> dim wave as qwave declare sub playme declare sub pauseme declare sub stopme declare sub altervol declare sub alterpos(position as long) wave.devicetype="mpegvideo" wave.open("c:\ms3\mybis\gx\runwaters.mp3") wave.onchange=alterpos create testme as qform caption="WAVER":center:width=500:height=450 create remark as qlabel left=20:top=250:caption="(press the pause or play button)" end create create remark2 as qlabel left=20:top=280:caption="Mode" end create create remark3 as qlabel left=20:top=310:caption="(info)" end create create playbtn as qbutton left=20:top=30:caption="Play":onclick=playme end create create pausebtn as qbutton left=120:top=30:caption="Pause":enabled=0 onclick=pauseme end create create stopbtn as qbutton left=220:top=30:caption="Stop":enabled=0 onclick=stopme end create create gauge as qgauge left=10:top=2:height=12:width=350 end create create volm as qtrackbar max=100:left=20:top=250:width=300:position=100 onchange=altervol end create end create SetWindowLong(testme.Handle, -8, 0) SetWindowLong(Application.Handle, -8, testme.Handle) </pre>	<p>Volume. The user may control the volume of the file being played.</p> <p>Sub program to alter volume.</p> <p>Track bar for the user to control the volume.</p>



SNO	CODING	EXPLANATION
	<pre> gauge.position=0:gauge.max=wave.lenght testme.showmodal sub playme wave.play:playbtn.enabled=0:pausebtn.enabled=1:stopbtn.enabled=1 if wave.state=1 then remark2.caption="Playing" remark3.caption="Bits: "+str\$(wave.bits)+chr\$(13) remark3.caption=remark3.caption+"Freq: "+str\$(wave.frequency)+chr\$(13) if wave.mode=1 then remark3.caption=remark3.caption+"Mono Mode" end if if wave.mode=2 then remark3.caption=remark3.caption+"Stereo Mode" end if end sub sub pauseme wave.pause:playbtn.enabled=1:pausebtn.enabled=0 if wave.state=2 then remark2.caption="Paused" end sub sub stopme wave.stop:playbtn.enabled=1:pausebtn.enabled=0:stopbtn.enabled=0 if wave.state=3 then remark2.caption="Stopped" end sub sub alterpos(position as long) remark.caption="Playing: "+str\$(fix(position/1000))+"/"+str\$(fix(wave.lenght/1000)) if wave.state=4 then remark2.caption="Recording" gauge.position=position end sub sub altervol wave.volume=volm.position end sub </pre>	<p>Adjust volume according to position of track bar [0-100]</p>
9	<pre> ... Wave.close Wave.open ("...") ... </pre>	<p>Before loading another file, you need to close the file (OPEN)</p>
10	<pre> \$TYPECHECK ON \$include <rapidq2.inc> \$include <qwave.inc> dim wave as qwave declare sub playme declare sub pauseme declare sub stopme declare sub altervol declare sub alterpos(position as long) declare sub startrec declare sub stoprect wave.open("c:\ms3\mybis\gx\ e28logo.wav") wave.onchange=alterpos create testme as qform caption="WAVER":center:width=500:height=450 create recbtn as qbutton left=20:top=100:caption="Rec" onclick=startrec end create create recstop as qbutton left=120:top=100:caption="Stop Rec" onclick=stoprect </pre>	<p>Recording & Stop recording.</p> <p>Sub program for recording and stop recording.</p> <p>Buttons to record.</p> <p>Button to stop recording.</p>



SNO	CODING	EXPLANATION
	<pre> end create create remark as qlabel left=20:top=250:caption="(press the pause or play button)" end create create remark2 as qlabel left=20:top=280:caption="Mode" end create create remark3 as qlabel left=20:top=310:caption="(info)" end create create playbtn as qbutton left=20:top=30:caption="Play":onclick=playme end create create pausebtn as qbutton left=120:top=30:caption="Pause":enabled=0 onclick=pauseme end create create stopbtn as qbutton left=220:top=30:caption="Stop":enabled=0 onclick=stopme end create create gauge as qgauge left=10:top=2:height=12:width=350 end create create volm as qtrackbar max=100:left=20:top=265:width=300:position=100 onchange=altervol end create end create SetWindowLong(testme.Handle, -8, 0) SetWindowLong(Application.Handle, -8, testme.Handle) gauge.position=0:gauge.max=wave.lenght testme.showmodal sub playme gauge.position=0:gauge.max=wave.lenght wave.play:playbtn.enabled=0:pausebtn.enabled=1:stopbtn.enabled=1 if wave.state=1 then remark2.caption="Playing" remark3.caption="Bits: "+str\$(wave.bits)+chr\$(13) remark3.caption=remark3.caption+"Freq: "+str\$(wave.frequence)+chr\$(13) if wave.mode=1 then remark3.caption=remark3.caption+"Mono Mode" end if if wave.mode=2 then remark3.caption=remark3.caption+"Stereo Mode" end if end sub sub pauseme wave.pause:playbtn.enabled=1:pausebtn.enabled=0 if wave.state=2 then remark2.caption="Paused" end sub sub stopme wave.stop:playbtn.enabled=1:pausebtn.enabled=0:stopbtn.enabled=0 if wave.state=3 then remark2.caption="Stopped" end sub sub alterpos(position as long) remark.caption="Playing: "+str\$(fix(position/1000))+"/"+str\$(fix(wave.lenght/1000)) if wave.state=3 then playbtn.enabled=1:pausebtn.enabled=0:stopbtn.enabled=0 </pre>	<p>Reset position when reloading a sound file.</p> <p>Reset position when reaching end of sound.</p>



SNO	CODING	EXPLANATION
	<pre> if wave.state=4 then remark2.caption="Recording" gauge.position=position end sub sub altervol wave.volume=volm.position end sub sub startrec wave.close wave.new wave.lenght=60000 gauge.position=0:gauge.max=wave.lenght wave.bits=16 wave.frequency=44100 wave.mode=2 wave.record recbtn.enabled=0:stopbtn.enabled=0 playbtn.enabled=0:recstop.enabled=1 end sub sub stoprect wave.stop wave.save "c:\ms3\~common\qwfile.wav" wave.close wave.open "c:\ms3\~common\qwfile.wav" gauge.position=0:gauge.max=wave.lenght recbtn.enabled=1:stopbtn.enabled=0 playbtn.enabled=1:recstop.enabled=0 end sub </pre>	<p>Close the current file. Start a new file (if not then overwrite on current open file). You must indicate the time required to record. If you stop earlier, then only the recorded time will be stored. Reset the gauge. Set the recording settings (bits, freq, mode). Start recording.</p> <p>Stop recording. Store the wave file at the indicate position. Close the file and reload the stored file.</p>
11	<pre> ... Wave.stop Wave.delete(3000,9000) ... </pre>	<p>Delete partial area as indicated.</p> <p>Stop the sequence. Delete from seconds 3 - 9.</p>

16.7 MP3

This element will specifically play MP3 files.

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> #include <mp3q.inc> declare sub playmp3 CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480:Center create playbtn as qbutton left=20:top=50:caption="Play":onclick=playmp3 end create END CREATE mp3file="c:\ms3\mybis\gx\runwaters.mp3" SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal sub playmp3 end sub </pre>	<p>Include this element. Sub program to play the MP3 sound file.</p> <p>Button to be pressed to play the MP3 sound file.</p> <p>Load the MP3 file.</p> <p>Sub program to play the sound (not playing yet)</p>

SNO	CODING	EXPLANATION
2	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <mp3q.inc> declare sub playmp3 CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480:Center create playbtn as qbutton left=20:top=50:caption="Play":onclick=playmp3 end create END CREATE mp3file="c:\ms3\mybis\gx\runwaters.mp3" SetWindowLong[Form.Handle, -8, 0] SetWindowLong[Application.Handle, -8, Form.Handle] Form.ShowModal sub playmp3 cmd="Open " & chr\$(34) & mp3file & chr\$(34) & " Alias Rapid-MP3" ret=mcisendstring(cmd,0,0,0) ret=mcisendstring(play,0,0,) end sub </pre>	<p>Playing the loaded file.</p>
3	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <mp3q.inc> declare sub playmp3 declare sub pausemp3 CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480:Center create playbtn as qbutton left=20:top=50:caption="Play":onclick=playmp3 end create create pausebtn as qbutton left=20:top=90:caption="Pause":onclick=pausemp3 end create END CREATE mp3file="c:\ms3\mybis\gx\runwaters.mp3" SetWindowLong[Form.Handle, -8, 0] SetWindowLong[Application.Handle, -8, Form.Handle] Form.ShowModal sub playmp3 cmd="Open " & chr\$(34) & mp3file & chr\$(34) & " Alias Rapid-MP3" ret=mcisendstring(cmd,0,0,0) ret=mcisendstring(play,0,0,) end sub sub pausemp3 mcisendstring("Stop Rapid-Mp3",0,0,0) end sub </pre>	<p>Pause Mode</p> <p>Sub program to pause sequence.</p> <p>Button to pause sequence.</p> <p>Command to pause sequence.</p>
4	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <mp3q.inc> declare sub playmp3 declare sub pausemp3 declare sub stopmp3 CREATE Form AS QFORM Caption = "Form":Width = 640:Height = 480:Center create playbtn as qbutton left=20:top=50:caption="Play":onclick=playmp3 end create create pausebtn as qbutton left=20:top=90:caption="Pause":onclick=pausemp3 </pre>	<p>Stop sequence (restart from beginning when Play is pressed again).</p>



SNO	CODING	EXPLANATION
	<pre> end create create stopbtn as qbutton left=20:top=140:caption="Stop":onclick=stopmp3 end create END CREATE mp3file="c:\ms3\mybis\gx\runwaters.mp3" SetWindowLong[Form.Handle, -8, 0] SetWindowLong[Application.Handle, -8, Form.Handle] Form.ShowDialog sub playmp3 cmd="Open " & chr\$(34) & mp3file & chr\$(34) & " Alias Rapid-MP3" ret=mcisendstring(cmd,0,0,0) ret=mcisendstring(play,0,0,) end sub sub pausemp3 mcisendstring("Stop Rapid-Mp3",0,0,0) end sub sub stopmp3 mcisendstring("Stop Rapid-Mp3",0,0,0) mcisendstring("Close Rapid-Mp3",0,0,0) end sub </pre>	<p>Button to be pressed to stop sequence.</p> <p>Stop sequence. Reset sequence to beginning.</p>

16.8 VOICE

This component will enable the computer to speak.

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> declare sub sayit Dim sapi as QOLEObject sapi.CreateObject("SAPI.SpVoice") CREATE Form AS QFORMex Caption = "Form":Width = 640:Height = 480:Center create playbtn as qbutton left=20:top=50:caption="Say":onclick=sayit end create create wordstosay as Qedit left=20:top=10:width=500: text="Are you impressed?" end create END CREATE sapi.speak("Welcome Andre", 1) SetWindowLong[Form.Handle, -8, 0] SetWindowLong[Application.Handle, -8, Form.Handle] Form.ShowDialog sub sayit sapi.speak[wordstosay.text, 1] end sub </pre>	<p>Activate the speech system.</p> <p>Button to be clicked to say what is typed in the EDIT field.</p> <p>Type something here – this will be spoken.</p> <p>Welcome speech.</p> <p>Speak what is in the EDIT field.</p>

TO STOP the voice;

TO ALTER the SPEED of the voice [-10 ... 0 ... 10];

sapi.skip "Sentence", 10

sapi.rate=2



16.9 CD Player

This component will play CD's that is loaded into the DVD/CD player.

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <qcdaudio.inc> Dim cd as Qcdaudio CREATE Form AS QFORMex Caption = "CD Player":Width = 640:Height = 480 color=32896:Center create btplay as qbutton left=60:top=50:width=50:caption="Play" end create create btback as qbutton left=10:top=50:width=50:caption="<<" showhint=1:hint="rewind" end create create btpause as qbutton left=110:top=50:width=50:caption=" " showhint=1:hint="pause" end create create btforward as qbutton left=160:top=50:width=50:caption=">>" showhint=1:hint="fast forward" end create create btstop as qbutton left=210:top=50:width=50:caption="Stop" end create create bteject as qbutton left=270:top=50:width=50:caption="<" showhint=1:hint="eject CD" end create CREATE Infos AS QSTATUSBAR SizeGrip=false AddPanels "Track:", "", "" Panel(0).width=60:Panel(1).width=100 END CREATE END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal </pre>	<p>Include the CD player element. Establish the CD element.</p> <p>Play button.</p> <p>Rewind button.</p> <p>Pause button.</p> <p>Fast forward button.</p> <p>Stop button.</p> <p>Eject button.</p> <p>Status bar to display the CD information (track nr, track time, cd time).</p>
2	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <qcdaudio.inc> Dim cd as Qcdaudio declare sub open CREATE Form AS QFORMex Caption = "CD Player":Width = 640:Height = 480 color=32896:Center create btplay as qbutton left=60:top=50:width=50:caption="Play" end create create btback as qbutton left=10:top=50:width=50:caption="<<" showhint=1:hint="rewind" end create </pre>	<p>Sub program to load the CD into memory.</p>

SNO	CODING	EXPLANATION
	<pre> create btpause as qbutton left=110:top=50:width=50:caption=" " showhint=1:hint="pause" end create create btforward as qbutton left=160:top=50:width=50:caption=">>" showhint=1:hint="fast forward" end create create bstop as qbutton left=210:top=50:width=50:caption="Stop" end create create bteject as qbutton left=270:top=50:width=50:caption="<<" showhint=1:hint="eject CD" end create CREATE Infos AS QSTATUSBAR SizeGrip=false AddPanels "Track:", "", "" Panel(0).width=60:Panel(1).width=100 END CREATE END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) open Form.ShowModal Sub Open if cd.open then infos.panel(0).caption="Track:"+str\$(cd.tracknumber) infos.panel(1).caption=cd.time btback.enabled=true:btforward.enabled=true btpause.enabled=true:bstop.enabled=true else showmessage cd.error end if End Sub </pre>	<p>Load the CD on bootup.</p> <p>If a CD is present, then load the CD information. Track numbers. Track Time. Enable buttons to play.</p> <p>If no CD then display error message.</p>
3	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <qcdaudio.inc> Dim cd as Qcdaudio declare sub open declare sub play declare sub stop declare sub pause declare sub forward declare sub back CREATE Form AS QFORMex Caption = "CD Player":Width = 640:Height = 480 color=32896:Center create btplay as qbutton left=60:top=50:width=50:caption="Play":onclick=play end create create btback as qbutton left=10:top=50:width=50:caption="<<":onclick=back showhint=1:hint="rewind" end create create btpause as qbutton left=110:top=50:width=50:caption=" ":onclick=pause showhint=1:hint="pause" end create </pre>	<p>Play, Pause, Rewind, Fast Forward and Stop</p>



SNO	CODING	EXPLANATION
	<pre> create btforward as qbutton left=160;top=50;width=50:caption=">>":onclick=forward showhint=1:hint="fast forward" end create create btstop as qbutton left=210;top=50;width=50:caption="Stop":onclick=stop end create create bteject as qbutton left=270;top=50;width=50:caption="<<" showhint=1:hint="eject CD" end create CREATE Infos AS QSTATUSBAR SizeGrip=false AddPanels "Track:", "", "" Panel(0).width=60:Panel(1).width=100 END CREATE END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) open Form.ShowModal Sub Open if cd.open then infos.panel(0).caption="Track:"+str\$(cd.tracknumber) infos.panel(1).caption=cd.time btback.enabled=true:btforward.enabled=true btpause.enabled=true:btstop.enabled=true else showmessage cd.error end if End Sub Sub Play if cd.audioOpen=false then if cd.open then infos.panel(0).caption="Track:"+str\$(cd.tracknumber) infos.panel(1).caption=cd.time else showmessage cd.error end if end if cd.play End Sub sub stop cd.stop end sub sub pause cd.pause end sub Sub Back cd.currentTrack=cd.currentTrack-1 infos.panel(0).caption="Track:"+str\$(cd.currentTrack) infos.panel(2).caption=cd.tracktime: if cd.state=CD_PLAY then cd.play End Sub Sub Forward cd.currentTrack=cd.currentTrack+1 infos.panel(0).caption="Track:"+str\$(cd.currentTrack) infos.panel(2).caption=cd.tracktime if cd.state=CD_PLAY then cd.play End Sub </pre>	<p>Stop playing</p> <p>Pause playing</p> <p>Go to previous track Reset status information Continue playing</p> <p>Go to next track track Reset status information Continue playing</p>



SNO	CODING	EXPLANATION
4	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <qcdaudio.inc> Dim cd as Qcdaudio declare sub open declare sub play declare sub stop declare sub pause declare sub forward declare sub back Declare Sub PostimerOnTimer(track as integer,time as string) cd.OnChange=PostimerOnTimer:cd.timer.interval=200 CREATE Form AS QFORMex Caption = "CD Player":Width = 400:Height = 200 color=32896:Center create btplay as qbutton left=60:top=50:width=50:caption="Play":onclick=play end create create btback as qbutton left=10:top=50:width=50:caption="<<":onclick=back showhint=1:hint="rewind" end create create btpause as qbutton left=110:top=50:width=50:caption=" ":onclick=pause showhint=1:hint="pause" end create create btforward as qbutton left=160:top=50:width=50:caption=">>":onclick=forward showhint=1:hint="fast forward" end create create btstop as qbutton left=210:top=50:width=50:caption="Stop":onclick=stop end create create bteject as qbutton left=270:top=50:width=50:caption=">" showhint=1:hint="eject CD" end create CREATE Infos AS GSTATUSBAR SizeGrip=false AddPanels "Track:", "", "" Panel(0).width=60:Panel(1).width=100 END CREATE END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) open Form.ShowModal Sub Open if cd.open then infos.panel(0).caption="Track:"+str\$(cd.tracknumber) infos.panel(1).caption=cd.time btback.enabled=true:btforward.enabled=true btpause.enabled=true:btstop.enabled=true else showmessage cd.error end if End Sub </pre>	<p>Monitor frame position (playing position)</p> <p>Sub program to determine playing position. Interval of .2 seconds to update position.</p>

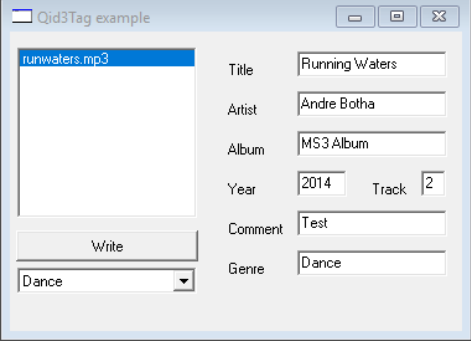


SNO	CODING	EXPLANATION
	<pre> Sub Play if cd.audioOpen=false then if cd.open then infos.panel(0).caption="Track:"+str\$(cd.tracknumber) infos.panel(1).caption=cd.time else showmessage cd.error end if end if cd.play End Sub sub stop cd.stop end sub sub pause cd.pause end sub Sub Back cd.currentTrack=cd.currentTrack-1 infos.panel(0).caption="Track:"+str\$(cd.currentTrack) infos.panel(2).caption=cd.tracktime if cd.state=CD_PLAY then cd.play End Sub Sub Forward cd.currentTrack=cd.currentTrack+1 infos.panel(0).caption="Track:"+str\$(cd.currentTrack) infos.panel(2).caption=cd.tracktime if cd.state=CD_PLAY then cd.play End Sub Sub PostimerOnTimer(track as integer,time as string) infos.panel(0).caption="Track:"+str\$(track) infos.panel(2).caption=time End Sub </pre>	<p>Track number Update time (position)</p>
5	<pre> ... CD.Eject ... Or ... CD.Close ... </pre>	<p>Eject / Closing of CD player</p>

16.10 QIDTAGS

MP3 Files are able to contain ID Tags (information regarding the file/song). You may select a MP3 file and then load and view the information or alter the information as you wish.



SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON #include "rapidq2.inc" #include "Qid3tag.inc" dim mp3 as qid3tag declare sub ReadClickedFile declare sub UpdateTags CREATE Form AS QFORM Caption = "Qid3Tag example" Width = 381 Height = 280 Center CREATE Label1 AS QLABEL Caption = "Title" Left = 176 Top = 24 Transparent = 1 END CREATE CREATE Label2 AS QLABEL Caption = "Artist" Left = 176 Top = 56 Transparent = 1 END CREATE CREATE Label3 AS QLABEL Caption = "Album" Left = 176 Top = 88 Transparent = 1 END CREATE CREATE Label4 AS QLABEL Caption = "Year" Left = 176 Top = 120 Transparent = 1 END CREATE create tlabel as qlabel caption = "Track" left = 292 top = 120 transparent = 1 end create CREATE Label5 AS QLABEL Caption = "Comment" Left = 176 Top = 152 Width = 48 Transparent = 1 END CREATE CREATE Label6 AS QLABEL Caption = "Genre" Left = 176 Top = 184 Transparent = 1 END CREATE CREATE FileListBox AS QFILELISTBOX Left = 6 Top = 13 Height = 137 Directory = "c:\ms3\mybis\gx\" </pre>	



SNO	CODING	EXPLANATION
	<pre> mask="*.mp3" onclick = ReadClickedFile END CREATE CREATE ed_title AS QEDIT Left = 232:Top = 16 END CREATE CREATE ed_artist AS QEDIT Left = 232:Top = 48 END CREATE CREATE ed_album AS QEDIT Left = 232:Top = 80 END CREATE CREATE ed_year AS QEDIT Left = 232:Top = 112:width = 40 END CREATE create ed_track as qedit left = 332:top = 112:width = 20 end create CREATE ed_comment AS QEDIT Left = 232:Top = 144 END CREATE CREATE ed_genre AS QEDIT Left = 232 Top = 176 END CREATE CREATE Button1 AS QBUTTON Caption = "Write" Left = 5 Top = 160 Width = 147 onclick = UpdateTags END CREATE END CREATE mp3.genrescb.parent = form.mp3.genrescb.top = 190 mp3.genrescb.left = 6:mp3.loadlist Form.ShowModal sub ReadClickedFile mp3.filename = Filelistbox.filename mp3.readtags mp3.genrescb.itemindex = mp3.genreid ed_title.text = mp3.title:ed_artist.text = mp3.artist ed_album.text = mp3.album:ed_year.text = mp3.year ed_comment.text = mp3.comment ed_track.text = str\$(mp3.track) ed_genre.text = mp3.genre end sub Sub UpdateTags mp3.title = ed_title.text:mp3.artist = ed_artist.text mp3.album = ed_album.text:mp3.year = ed_year.text mp3.comment = ed_comment.text mp3.track = val(ed_track.text) mp3.genreid = mp3.genrescb.itemindex mp3.writetags end sub </pre>	<p>Only MP3 files to be displayed.</p> <p>This will alter the information as entered within the fields. Select a file, update and click this button.</p> <p>Assign element to FORM.</p> <p>Load the selected file. Read the file's ID tags.</p>

