

MODULE 23: VIDEO

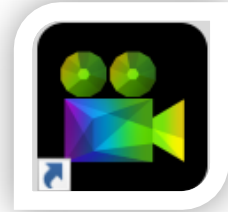
On Completion of this module you will be able to add videos to your software.

Subject Outcome 1: Introduction

Subject Outcome 2: QVideo

Subject Outcome 3: QAVI and AVI recorder





23.1 INTRODUCTION

Sometimes still (photos, image, gif, etc.) just isn't sufficient and video files are required. Be it for presentation, training, instructional or just watching videos, you should be able to include it into your program for the user to enjoy the full capability of a true program.

The disadvantage to playing video files is 3rd party Video activations. Certain formats such as MP4, etc. must be registered with 3rd party owner, so if your computer does not have those registrations, it will not play. This makes the following elements Windows reliant.

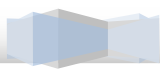
The only two formats guaranteed to be play are WMV and AVI formats. This means that if you are using a tutorial video within your program, convert the video file to WMV or AVI. A wonderful FREE convertor program is MMCONVERT or even the installed Windows Movie Maker program. It will be able to open and edit the video you require and then resave it in the WMV or AVI format that you wish to use (be careful as AVI is very large files).

23.2 QVIDEO

QVideo is a media component used to play a video file, such as AVI, MPEG, etc. using Windows multimedia support. Remember it is Windows Media player reliant.

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <qvideo.inc> dim openfile as qopendialog openfile.filter="Video files .avi;*.wmv;*.mpg;*.mp4 All Files *.*" openfile.filterindex=0 declare Sub PostimerOnTimer(Position as long,timePos as long) dim totaltime as string dim min as integer dim sec as integer dim seconds as long Dim video as QVideo video.OnChange=PostimerOnTimer video.timer.interval=200 declare sub openvideo declare sub playvideo CREATE Form AS QFORMex Caption = "Videos":Width = 900:Height = 550:Center create mainmenu as qmainmenu create menu1 as qmenuItem caption="Open":onclick=openvideo end create create menu2 as qmenuItem caption="PLAY":onclick=playvideo end create </pre>	<p>Include the QVIDEO element. To open the video file.</p> <p>This will monitor the position of play. The variables required by the element.</p> <p>Establish the element. Monitor play motion (position). Check every .200 second for play motion.</p> <p>The sub programs to open a video file and play the video.</p> <p>Menu button to open a file.</p> <p>Menu button to play the opened file.</p>

SNO	CODING	EXPLANATION
	<pre> create menu3 as qmnuitem caption="Pause" end create create menu4 as qmnuitem caption="Stop" end create create menu5 as qmnuitem caption="(duration)":enabled=0 end create create menu6 as qmnuitem caption="(size)":enabled=0 end create end create create posbar as qtrackbar left=10:top=2:height=8:width=300:max=1000:min=0 end create create volume as qtrackbar orientation=1:position=0 left=10:width=25:top=20:height=200 max=100:min=0:showhint=1:hint="volume" end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) video.parent=form.handle Form.ShowDialog sub openvideo if openfile.execute then if video.fileopen=1 then video.close end if form.caption="VIDEO: "+openfile.filename video.open (openfile.filename) video.show seconds=video.LenghtTime posbar.max=seconds min=seconds/60 sec=seconds mod 60 if sec<10 then totaltime=str\$(min)+":0"+str\$(sec) else totaltime=str\$(min)+":"+str\$(sec) end if menu5.caption=totaltime menu6.caption=str\$(video.imgwidth)+" / "+str\$(video.imgheight) form.width=video.imgwidth+68:form.height=video.imgheight+80 form.center video.left=50:video.top=20 end if end sub sub playvideo if video.fileopen=1 then video.play end if end sub Sub PostimerOnTimer(Position as long,timePos as long) posbar.position=timepos end sub </pre>	<p>Pause button on the play.</p> <p>Stop the video playing.</p> <p>Indicate the duration of the opened video file.</p> <p>Indicate the screen resolution (x and y) of the video.</p> <p>Display and control the position of the video file playing.</p> <p>Control the video's volume.</p> <p>Assign the video to the FORM. If you used a PANEL then you would assign it to that PANEL.</p> <p>Opening the video file. Use dialog box. Check if a video file is already open before opening the new file; if so then close that file. Display the video name as Form's caption. OPEN the video file. Display the video file (in pause mode). Determine the length of the video and use some basic mathematics to determine the minutes and seconds.</p> <p>Display video screen size. Resize the FORM to fit the video. Center the newly formed FORM. Indicate the position of the VIDEO screen on the FORM.</p> <p>PLAY the video if a video file has been opened.</p> <p>Determine and indicate the position of the playing video - use the track-bar to display position.</p>



SNO	CODING	EXPLANATION
2	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> \$include <qvideo.inc> dim openfile as qopendialog openfile.filter="Video files .avi;*.wmv;*.mpg;*.mp4 All Files *.*" openfile.filterindex=0 declare Sub PostimerOnTimer(Position as long,timePos as long) dim totaltime as string dim c as double dim min as integer dim sec as integer dim seconds as long Dim video as QVideo video.OnChange=PostimerOnTimer video.timer.interval=200 declare sub openvideo declare sub playvideo declare sub pausevideo declare sub stopvideo declare sub updatenewposition declare sub updatevolumneknob CREATE Form AS QFORMex Caption = "Videos":Width = 900:Height = 550:Center create mainmenu as qmainmenu create menu1 as qmenuitem caption="Open":onclick=openvideo end create create menu2 as qmenuitem caption="PLAY":onclick=playvideo end create create menu3 as qmenuitem caption="Pause":onclick=pausevideo end create create menu4 as qmenuitem caption="Stop":onclick=stopvideo end create create menu5 as qmenuitem caption="{duration}":enabled=0 end create create menu6 as qmenuitem caption="{size}":enabled=0 end create end create create posbar as qtrackbar left=10:top=2:height=8:width=300:max=1000:min=0 onchange=updatenewposition end create create volume as qtrackbar orientation=1:position=0 left=10:width=25:top=20:height=200 max=100:min=0:showhint=1:hint="volume" onchange=updatevolumneknob end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) video.parent=form.handle Form.ShowModal sub openvideo if openfile.execute then </pre>	<p>Now to code the volume control, pause and stop button. We must also control the position of the video.</p> <p>To pause video sub program. Stop video sub program. Update video playing position. Update video volume.</p> <p>Button to pause video.</p> <p>Button to stop video.</p> <p>When the user drags the track-bar icon for the video's position.</p> <p>When the user drag the track bar icon for the volume.</p>



SNO	CODING	EXPLANATION
	<pre> if video.fileopen=1 then video.close end if form.caption="VIDEO: "+openfile.filename video.open (openfile.filename) video.show seconds=video.LenghtTime posbar.max=video.Lenght min=seconds/60 sec=seconds mod 60 if sec<10 then totaltime=str\$(min)+":0"+str\$(sec) else totaltime=str\$(min)+":"+str\$(sec) end if menu5.caption=totaltime menu6.caption=str\$(video.imgwidth)+" / "+str\$(video.imgheight) form.width=video.imgwidth+68:form.height=video.imgheight+80 form.center video.left=50:video.top=20 end if end sub sub playvideo if video.fileopen=1 then video.play end if end sub Sub PostimerOnTimer(Position as long,timePos as long) posbar.position=position end sub sub pausevideo video.pause end sub sub stopvideo video.stop:posbar.position=0 end sub sub updatenewposition if video.state<>VD_PLAY then video.CurrentFrame=posbar.position end if end sub sub updatevolumneknob video.volume=100-volume.position end sub </pre>	<p>I altered the track bar MAX value to the frame count of the video and not time [seconds]. It is easier to control the position bar this way.</p> <p>Pause the video.</p> <p>Stop and reset position of the video.</p> <p>If the video is in stop or pause mode then alters the video's position using the frame position. You cannot alter the position in playing mode.</p> <p>Alter the volume [100- is due to the reverse position of the volume trackbar].</p>

23.3 QAVI

QAVI is a media component used to determine properties and even record a video file using BMP images - AVI format only. We will firstly load an AVI file and display it's properties. Thereafter we will create a new AVI file and record some BMP images to it -



you may either create individual BMP images or load BMP images one at a time and record an AVI from it – **important to note is that all the BMP images must be the same size!!!**.

This is actually NOT a video player, but a frame editor – if you are now wondering how to make a frame shot (picture) of the QVIDEO player (when in pause mode), simply code a screen shot grabber (module 11) on the set coordinates of the QVIDEO screen (left, right, top, bottom – QRECT). We will now code the structure of the program, then the frame viewer and lastly coding a group of BMP images as an AVI video (no sound recording – only images).

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> #include <qavi.inc> dim myavi as qavi myavi.showerror=1 dim openfile as qpendialog openfile.filter="Video files*.avi All Files *.*" openfile.filterindex=0 dim totaltime as string dim i as single dim min as integer:dim sec as integer:dim seconds as long declare sub openvideo CREATE Form AS QFORMex Caption = "Videos AVI":Width = 900:Height = 550:Center create mainmenu as qmainmenu create menu1 as qmenuitem caption="Open":onclick=openvideo end create create menu3b as qmenuitem caption="(Duration)":enabled=0 end create create menu4 as qmenuitem caption="(FPS)":enabled=0 end create create menu5 as qmenuitem caption="(Frames)":enabled=0 end create create menu6 as qmenuitem caption="(size)":enabled=0 end create end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal sub openvideo if openfile.execute then form.caption="VIDEO: "+openfile.filename myavi.getavifileinfo(openfile.filename) menu4.caption="FPS: "+str\$(myavi.fps) menu5.caption="FRAMES: "+str\$(myavi.numframes) menu6.caption="SIZE: "+str\$(myavi.width)+" / "+str\$(myavi.height) form.clientwidth=myavi.width:form.clientheight=myavi.height </pre>	<p>Include the AVI element Establish the element. Display an error if ever found. The open dialog box to be used to call a file.</p> <p>Some variables required to display information.</p> <p>The sub program to call the AVI file.</p> <p>Open a file.</p> <p>The time duration of the AVI file.</p> <p>The frames per second.</p> <p>The total frames of the entire AVI file.</p> <p>The screen size of the AVI video.</p> <p>Open the file. Frames Per Second Total frames Screen resolution. Adjust form's size as AVI size.</p>



SNO	CODING	EXPLANATION
	<pre> if form.clientwidth<500 then form.clientwidth=500 if form.clientheight<240 then form.clientheight=240 seconds=floor[myavi.numframes/myavi.fps] min=seconds/60 sec=seconds mod 60 if sec<10 then totaltime=str\$(min)+"0"+str\$(sec) else totaltime=str\$(min)+"."+str\$(sec) end if menu3b.caption="Duration: "+totaltime form.center end if end sub </pre>	<p>Determine total duration in minutes and seconds.</p>

With the next lecture we will record an AVI file with some BMP images.

SNO	CODING	EXPLANATION
1	<pre> \$TYPECHECK ON \$INCLUDE <rapidq2.inc> \$INCLUDE <QAVI.inc> DECLARE SUB PaintForm DIM i as integer CREATE TheBMP AS QBITMAP Width = 160:Height = 120 PixelFormat = pf24bit END CREATE CREATE Form AS QFORM ClientWidth = TheBMP.Width ClientHeight = TheBMP.Height OnPaint=PaintForm END CREATE CREATE myAVI AS QAVI ShowError = True NumFrames = 50 END CREATE myAVI.CreateAVIFile("c:\ms3\~common\AviTest.avi", TheBMP.Width, TheBMP.Height, 24, 10) showmessage "File created ..." Form.Show for i=1 to myAVI.NumFrames WITH TheBMP .FillRect(0,0, .width, .height, 0) .Line(0,0, .width*Rnd, .height*Rnd, &Hff00) .textout(0,0,"MS3 Systems",255,-1) END WITH myAVI.BMPtoAVIFile(TheBMP) Form.Repaint next i myAVI.CloseAVIFile Application.Terminate SUB PaintForm Form.Draw(0,0, TheBMP.BMP) END SUB </pre>	<p>Include the QAVI element</p> <p>Establish the QBITMAP that will contain the frame images. Only 24bit is accepted.</p> <p>Establish the QAVI element.</p> <p>Assign 50 frames.</p> <p>Create the AVI file, width, height, bit format and FPS).</p> <p>Loop some frames that are created individually with basic sketching methods and some text.</p> <p>Assign each frame individually. Update form QBITMAP image.</p> <p>Close the file to free memory. Stop the program and exit.</p> <p>Update the BMP image.</p>

