

Converting wav files to MP3

by -

[mike_ukmid](#)

There are many examples of LB sound recorders which record in wav format, a format which consumes huge hard disk space. There are also many freeware wav to MP3 utilities but it is possible to include wav to mp3 conversion in your LB programs using a free command line executable, lame.exe.

Lame.exe can be obtained from <http://www.rarewares.org/mp3-lame-bundle.php> by downloading Lame3.98.2.zip (549kB) and the zip file also contains all the information required to make use of lame.exe

See the various html files within the zip for details on setting sample rates, bitrates, etc. The demo is set to a bitrate of 160kB/s.

- Create a new folder and from the zip file extract only lame.exe to that folder.
- Download the source file from the link at the bottom of the page or start LB, copy/paste the code below and save as lb-lame.bas to the same folder.
- Run lb-lame.bas, locate a wav file, the newly generated mp3 file will be saved to the same folder as the wav file.
- Lame.exe is run 'hidden' but if you want to observe the program, change s.nShow.struct=_SW_HIDE to s.nShow.struct=_SW_NORMAL

Other LAME Options to re-encode existing mp3 files.

Apart from wav to mp3 conversion, lame.exe is also able to modify existing mp3 files, for example changing the bitrate or resampling or applying highpass/lowpass filters, etc. Switches.html, found inside the zip file, documents all the command line switches with examples of how to apply them. As an example, to change the bitrate (and subsequently reduce the file size) of an already encoded file which has say a bitrate of 160kb/s, to 80kb/s, run lame.exe with parameters '-b 80 source.mp3 dest.mp3'. If you have mp3 files which you would like to play in your LB programs, using native playwave command, lame.exe can convert them to wav too!

The code below could be added to most of the LB wave recorder demos, record sound as wav, convert to mp3 then immediately delete the wave file.

Enjoy!

```
'converting wave file to mp3 (M.Bradbury April 2009)
'locate a wav file & build mp3 file name
filedialog "Locate wav file ...","*.wav",wavfile$
if wavfile$<>"" then
wpath$=getPath$(wavfile$)
```

```
wfile$=getfilename$(wavfile$)
mp3file$=left$(wfile$,len(wfile$)-3)+"mp3"
'set up conversion params
'-b 96,112,128,160,192 see documentation for full bitrate range & other settings.
quality$="-b 160"
paramStr$=quality$+chr$(32)+wfile$+chr$(32)+mp3file$ '-h a.wav a.mp3
lamefile$="lame.exe"

struct s,_
cbSize as long,fMask as long,hwnd as ulong,_
lpVerb$ as ptr,lpFile$ as ptr,lpParameters$ as ptr ,_
lpDirectory$ as ptr,nShow as long,hInstApp as long,_
lpIDList as long,lpClass as long,hkeyClass as long,_
dwHotKey as long,hIcon as ulong,hProcess as ulong
'end struct

SEEMASKNOCLOSEPROCESS = 64
s.cbSize.struct=len(s.struct)
s.fMask.struct=SEEMASKNOCLOSEPROCESS
s.hwnd.struct=0
s.lpVerb$.struct="open"
s.lpFile$.struct=lamefile$
s.lpParameters$.struct=paramStr$
s.lpDirectory$.struct=wpath$
s.nShow.struct=_SW_HIDE
print "Converting wav to mp3"
print wavfile$;" @start time: ";time$()
print "Please wait ...."
calldll #shell32 , "ShellExecuteExA",_
s as struct,r as long

if r<>0 then
    hProcess=s.hProcess.struct
    else
        print "Error - return=";r
        end
    end if

waitResult=-1
while waitResult<>0
calldll #kernel32, "WaitForSingleObject",_
hProcess as ulong,0 as long,_
waitResult as long
scan
'reduce processor load
```

```
call dll #kernel32,"Sleep",50 as ulong, res as void
wend

print "File conversion finished"
print wpath$;mp3file$;" @end time: ";time$()
print "End"
end if
end

function getFilename$(filespec$)
    fileindex=len(filespec$)
    filelength=len(filespec$)
    while mid$(filespec$, fileindex,1)<>"\""
        fileindex=fileindex-1
        if fileindex=0 then exit while
        scan
    wend
    getFilename$=right$(filespec$,filelength-fileindex)
end function

function getPath$(filespec$)
    fileindex=len(filespec$)
    filelength=len(filespec$)
    while mid$(filespec$, fileindex,1)<>"\""
        fileindex=fileindex-1
        if fileindex=0 then exit while
        scan
    wend
    getPath$=left$(filespec$,fileindex)
end function
```

Source file:

[lb-lame.bas](#)

- [Details](#)
- [Download](#)
- 2 KB

Bare Bones MP3 recorder.

Source file:

[MP3recorder.bas](#)

- [Details](#)
- [Download](#)
- 5 KB

```
*****  
**  
***** MP3 recorder, mike_ukmid April 2009. Based on Alyces Bare ***  
**  
***** Bones Wave recorder from 'pre-  
historic' times. *****  
***** MP3 encoder 'lame.exe' must be located in the same folder ***  
**  
***** as this program. ***  
**  
***** See http://lbpe.wikispaces.com/WAVtoMP3 for download link ***  
**  
***** for lame.exe ***  
**  
***** Encoding is NOT done on the fly. A wav file is encoded to ***  
**  
***** MP3 when recording is terminated. Wav file is then deleted. ***  
**  
*****  
**  
nomainwin  
WindowWidth=210:WindowHeight=120  
button #1.rec, "Record", [record],UL,10,10,80,26  
button #1.stp, "Stop", [stop],UL,100,10,80,26  
statictext #1.st1, "Progress:",10,40,200,20  
statictext #1.st2, "",10,65,200,20  
open "MP3 Recorder" for window_nf as #1  
#1, "trapclose [quit]"  
#1, "font arial 8"  
bits=16 '8  
bitrate=32000 '11025 16000 22050 32000 44100
```

```
bytespersec=32000
chans=2
wait

[record]
  isWave=0
  filedialog "Save MP3 file as...", "*.mp3", mp3$
  if mp3$<>"" then
    set$=
    "set myRecording time f
    ormat ms bitspersample "+str$(bits)+" channels "+str$(chans)+_
    " samplespersec "+str$(bitrate)
    r$=mciSendString$("open new type waveaudio alias myRecording")
    r$=mciSendString$(set$)
    r$=mciSendString$(
    "set myRecording alignment 1
    bytespersec "+str$(bytespersec))'save in this format
    r$=mciSendString$("record myRecording") 'start the recording
    #1.st2, "Recording ..."
    end if
wait

[stop]
  r$=mciSendString$("stop myRecording")
  r$=mciSendString$("save myRecording tempWav.wav")
  r$=mciSendString$("close myRecording")
  isWave=1
  #1.st2, "Finished"
  timer 1000,[go]
  wait
  [go]
  timer 0
  #1.st2, "MP3 conversion ..."
  r=wav2mp3(mp3$)
  if r=0 then
    #1.st2, "MP3 conversion failed"
    else
      #1.st2, "Playing MP3 file ..."
      r$=mciSendString$("open "
+chr$(34)+mp3$+chr$(34)+" type MpegVideo alias mymp3")
      r$=mciSendString$("play mymp3 wait")
      r$=mciSendString$("close mymp3")
      #1.st2, "Finished ..."
  end if
wait
```

```
[quit]
  if isWave then kill "tempWav.wav"
  close #1
end

function wav2mp3(mp3$) 'encode wav file to mp3
  'set up conversion params
  '-b 96,112,128,160,192 see documentation for full bitrate range & other settings.
  paramStr$="-b 160 tempWav.wav"
  +chr$(32)+chr$(34)+mp3$+chr$(34) '-h a.wav a.mp3
  lamefile$=DefaultDir$+"\lame.exe"
  struct s,_
    cbSize as long,fMask as long,hwnd as ulong,_
    lpVerb$ as ptr,lpFile$ as ptr,lpParameters$ as ptr ,_
    lpDirectory$ as ptr,nShow as long,hInstApp as long,_
    lpIDList as long,lpClass as long,hkeyClass as long,_
    dwHotKey as long,hIcon as ulong,hProcess as ulong
  'end struct
  SEEMASKNOCLOSEPROCESS = 64
  s.cbSize.struct=len(s.struct)
  s.fMask.struct=SEEMASKNOCLOSEPROCESS
  s.hwnd.struct=0
  s.lpVerb$.struct="open"
  s.lpFile$.struct=lamefile$
  s.lpParameters$.struct=paramStr$
  s.lpDirectory$.struct=DefaultDir$
  s.nShow.struct=_SW_HIDE
  calldll #shell32 , "ShellExecuteExA",_
  s as struct,r as long
  if r<>0 then
    hProcess=s.hProcess.struct
    wav2mp3=1
  else
    wav2mp3=0
    exit function
  end if
  waitResult=-1
  while waitResult<>0
    calldll #kernel32, "WaitForSingleObject",_
    hProcess as ulong,0 as long,_
    waitResult as long
    scan
    calldll #kernel32,"Sleep",50 as ulong, res as void
  wend
end function
```

```
function mciSendString$(s$)
    'Buffer will contain a return string from
    'the function, if there is one.
    buffer$=space$(1024)+chr$(0)
    call dll #winmm,"mciSendStringA",s$ as ptr,buffer$ as ptr,_
    1028 as long, 0 as long, r as long
    'truncate returned string at null character
    buffer$=left$(buffer$, instr(buffer$, chr$(0)) - 1)
    if r>0 then
        mciSendString$="error"
    else
        mciSendString$=buffer$
    end if
end function

function getFilename$(filespec$)
    fileindex=len(filespec$)
    filelength=len(filespec$)
    while mid$(filespec$, fileindex,1)<>"\""
        fileindex=fileindex-1
        if fileindex=0 then exit while
    wend
    getFilename$=right$(filespec$,filelength-fileindex)
end function

function getPath$(filespec$)
    fileindex=len(filespec$)
    filelength=len(filespec$)
    while mid$(filespec$, fileindex,1)<>"\""
        fileindex=fileindex-1
        if fileindex=0 then exit while
    wend
    getPath$=left$(filespec$,fileindex)
end function
```