

## Get the length of a wave file

This is an alternative way to the way described on the original site. Thanks to Alyce who shared the *ReverseString\$()* function.

NOTICE: It seems like sometimes (about 5% or less) the length is bugged (much smaller than it should be). I didn't notice it was bigger than the original though). I am trying to fix this ASAP, the updated code will be on this site.

EDIT: Found the bug. The problem was, that the *dechex\$()* function didn't return the 0 before the other letter/number. It should work now!

```
function getWavLength(file$)
open file$ for binary as #wav
content$ = input$(#wav, 32)
close #wav

if left$(content$, 4) <> "RIFF" then
notice file$ + chr$(13) + "This file is not a valid wave-file!"
end if

size$ = mid$(content$, 5, 4)
'Get the 4 bytes which contain the size of the file
size$ = ReverseString$(size$)
'Reverse those bytes because they are Little-Endian
size = hexdec(tohex$(asc(size$)) + tohex$(asc(right$(size$, 3))) +
tohex$(asc(right$(size$, 2))) + tohex$(asc(right$(size$, 1))))
size = size - 36
'Subtract 36 bytes because the actual sound data starts at byte 44, but
't the size in the header has already 8 bytes subtracted

bitrate$ = mid$(content$, 29, 4)
bitrate$ = ReverseString$(bitrate$)
bitrate = hexdec(tohex$(asc(bitrate$)) + tohex$(asc(right$(bitrate$, 3))) +
tohex$(asc(right$(bitrate$, 2))) + tohex$(asc(right$(bitrate$, 1)))))

getWavLength = int(size / bitrate)

end function

function ReverseString$(string$)
for i = len(string$) to 1 step -1
ReverseString$=ReverseString$+mid$(string$,i,1)
next i
```

```
end function

function tohex$(val)
tohex$ = dechex$(val)
if len(tohex$) = 1 then
tohex$ = "0"; tohex$
end if
end function
```