_MEMSOUND

The **_MEMSOUND** function returns a _MEM value referring to a sound's raw data in memory using a designated sound handle created by the _SNDOPEN or _SNDNEW function.

Syntax

soundBlock = _MEMSOUND(*soundHandle&*, *channel&*)

Contents

Parameters

- The *soundBlock* _MEM type variable holds the read-only elements .OFFSET, .SIZE, .ELEMENTSIZE, .TYPE and .SOUND.
  - .OFFSET is the starting memory address of the sound sample data.
  - .SIZE is the size of the sample data in **bytes**
  - .ELEMENTSIZE will contain the number of **bytes-per-sample** the audio contains.
    - Can return 1 (8-bit mono), 2 (8-bit stereo), 2 (16-bit mono), 4 (16-bit stereo), 4 (32-bit mono) or 8 (32-bit stereo).
    - Use .TYPE to determine the data type of the sample data.
  - .TYPE will contain the data type of the sample data. See _MEM for details.
  - .SOUND will contain the same handle value as returned by the _SNDOPEN function.
- The second parameter *channel&* must be 0 (interleaved/mono; version 3.1 and up)

Description

- Use this function to obtain a pointer to the raw sound data in memory for direct access.
- Even if the memory pointer obtained by this fuction was already freed again using _MEMFREE, the respective Sound handle itself must still be freed using _SNDCLOSE when no longer required.
- If .SIZE returns 0, that means the data could not be accessed. It may happen if you try to access the right channel in a mono file or the format simply does not support accessing raw PCM samples.
- *channel&* - 1 (left channel/mono) or 2 (right channel; for stereo files) was supported on the old OpenAL backend. For the new miniaudio backend, this must be 0.

Availability

- **QB64 v1.5 and up**
- **QB64-PE all versions**

Examples

Example 1
Checking that a sound file is stereo.

```
OPTION _EXPLICIT

PRINT "Loading...";
DIM Song AS LONG
Song = _SNDOPEN("onward_ride1.flac") ' Replace file name with your sound file
IF Song < 1 THEN
    PRINT "Failed to load sound!"
    END
END IF
PRINT "Done!"

DIM Channels AS _UNSIGNED _BYTE
Channels = SndChannels(Song)

IF Channels = 2 THEN
    PRINT "This file is STEREO"
ELSEIF Channels = 1 THEN
    PRINT "This file is MONO"
ELSE
    PRINT "An error occurred."
END IF

_SNDCLOSE Song 'closing the sound releases the mem blocks

END


' This function returns the number of sound channels for a valid sound "handle"
' 2 = stereo, 1 = mono, 0 = error
FUNCTION SndChannels~%% (handle AS LONG)
    DIM SampleData AS _MEM

    SndChannels = 0 ' Assume failure

    ' Check if the sound is valid
    SampleData = _MEMSOUND(handle, 0)
    IF SampleData.SIZE = 0 THEN
        EXIT FUNCTION
    END IF

    ' Check the data type and then decide if the sound is stereo or mono
    IF SampleData.TYPE = 260 THEN ' 32-bit floating point
        IF SampleData.ELEMENTSIZE = 4 THEN
            SndChannels = 1
        ELSEIF SampleData.ELEMENTSIZE = 8 THEN
```

```
            SndChannels = 2
        END IF
    ELSEIF SampleData.TYPE = 132 THEN ' 32-bit integer
        IF SampleData.ELEMENTSIZE = 4 THEN
            SndChannels = 1
        ELSEIF SampleData.ELEMENTSIZE = 8 THEN
            SndChannels = 2
        END IF
    ELSEIF SampleData.TYPE = 130 THEN ' 16-bit integer
        IF SampleData.ELEMENTSIZE = 2 THEN
            SndChannels = 1
        ELSEIF SampleData.ELEMENTSIZE = 4 THEN
            SndChannels = 2
        END IF
    ELSEIF SampleData.TYPE = 1153 THEN ' 8-bit unsigned integer
        IF SampleData.ELEMENTSIZE = 1 THEN
            SndChannels = 1
        ELSEIF SampleData.ELEMENTSIZE = 2 THEN
            SndChannels = 2
        END IF
    ELSEIF SampleData.TYPE = 0 THEN ' This means this is an OpenAL sound handle
        DIM RightChannel AS _MEM
        RightChannel = _MEMSOUND(handle, 2)
        IF RightChannel.SIZE > 0 THEN
            SndChannels = 2
        ELSE
            SndChannels = 1
        END IF
    END IF
END FUNCTION
```

Example 2
Plotting a sound's waves.

```
DEFLNG A-Z
OPTION _EXPLICIT

SCREEN _NEWIMAGE(800, 327, 32)

PRINT "Loading...";
DIM Song AS LONG
Song = _SNDOPEN("OPL3 Groove.rad") ' Replace this with your (rad, mid, it, xm, s3m, mod, mp3, flac, ogg, wav) sound file
IF Song < 1 THEN
    PRINT "Failed to load song!"
    END
END IF
PRINT "Done!"

_SNDPLAY Song
```

```
DIM SampleData AS _MEM
SampleData = _MEMSOUND(Song, 0)
IF SampleData.SIZE = 0 THEN
    PRINT "Failed to access sound sample data."
    END
END IF

DIM x AS LONG, i AS _UNSIGNED _INTEGER64, sf AS SINGLE, si AS INTEGER
DIM sz AS _UNSIGNED _INTEGER64

sz = _CV(_UNSIGNED _INTEGER64, _MK$(_OFFSET, SampleData.ELEMENTSIZE)) ' sz is the total size of the sound in bytes

DO UNTIL _KEYHIT = 27 OR NOT _SNDPLAYING(Song) OR i + (_WIDTH * sz) > SampleData.SIZE
    CLS
    LOCATE 1, 1: PRINT i; "/"; SampleData.SIZE, "Frame Size ="; sz, "Data Type ="; SampleData.TYPE

    $CHECKING:OFF
    IF SampleData.TYPE = 130 THEN ' integer stereo or mono
        FOR x = 0 TO _WIDTH - 1
            si = _MEMGET(SampleData, SampleData.OFFSET + i + x * sz, INTEGER) 'get sound data
            LINE (x, _HEIGHT / 2)-STEP(0, 300 * si / 32768), _RGB32(0, 111, 0) 'plot wave
        NEXT
    ELSEIF SampleData.TYPE = 260 THEN ' floating point stereo or mono
        FOR x = 0 TO _WIDTH - 1
            sf = _MEMGET(SampleData, SampleData.OFFSET + i + x * sz, SINGLE) 'get sound data
            LINE (x, _HEIGHT / 2)-STEP(0, sf * 300), _RGB32(0, 111, 0) 'plot wave
        NEXT
    ELSEIF sz = 2 AND SampleData.TYPE = 0 THEN ' integer mono (QB64 OpenAL stuff)
        FOR x = 0 TO _WIDTH - 1
            si = _MEMGET(SampleData, SampleData.OFFSET + i + x * sz, INTEGER) 'get sound data
            LINE (x, _HEIGHT / 2)-STEP(0, 300 * si / 32768), _RGB32(0, 111, 0) 'plot wave
        NEXT
    END IF
    $CHECKING:ON

    _DISPLAY
    _LIMIT 60

    i = FIX(_SNDGETPOS(Song) * _SNDRATE) * sz ' Calculate the new sample frame position
LOOP

_SNDCLOSE Song 'closing the sound releases the mem blocks
_AUTODISPLAY
END
```

See also

- _MEM, _MEMFREE
- _MEMPUT, _MEMGET, _MEMGET (function)
- _SNDOPEN, _SNDNEW, _SNDCLOSE, _SNDRAW
- _SNDRATE

Navigation:
Main Page with Articles and Tutorials
Keyword Reference - Alphabetical
Keyword Reference - By usage