

QBXIOL
Input/Output Low-Level Library
for
QuickBASIC 4.x and BASIC 7.x

by
Cornel Huth

31-May-1991

The QBXIOIOL software module is a DOS device input/output enhancement to Microsoft QuickBASIC 4.x and BASIC 7.x compilers. Its primary purpose is to provide fast I/O support for my QBXSBC SoundBlaster/AdLib music card software library but it can also be used separately.

Features of QBXIOIOL are:

- 1) Fast DOS I/O, written in fast and compact assembly code.
- 2) Critical error handling.
- 3) Unlimited file read/write size.
- 4) Read/write from any byte position within file or continue from last position.
- 5) Create files with specific file attributes.
- 6) Open files with specific access and sharing modes.

This software package is copyrighted material. You may use it for non-commercial work only. If you are going to be using QBXIOIOL commercially, or in any work that is to be distributed in any manner, you must register by purchasing a license for the QBXSBC package.

- QBXSBBC, the package \$19.95
- includes single-user licenses for:
- 1) QBXIOIOL, fast I/O DOS file module
 - 2) QBXCTV, digitized voice I/O module for SoundBlaster
 - 3) QBXFMI, interface module to the resident FM driver for the SoundBlaster and AdLib music cards. Includes QBXFMI.BAS and resident driver SB-SOUND.COM for the SoundBlaster. - also includes:
 - 4) Useful sample programs in QB demonstrating how to access AdLib instrument BNK files, voice files, and card detection.
 - 5) Programmer documentation

To order see the ORDER.FRM file.

LICENSE AGREEMENT - REGISTERED VERSIONS ONLY

This is a legal agreement between you, the end user, and Cornel Huth. By using this software, you are agreeing to be bound by the terms of this agreement.

SOFTWARE LICENSE

1. GRANT OF LICENSE. Cornel Huth grants to you the right to use one copy of the SOFTWARE on a single terminal connected to a single computer (i.e., with a single CPU). You may not network the SOFTWARE or otherwise use it on more than one computer or computer terminal at a time.
2. COPYRIGHT. The SOFTWARE is owned by Cornel Huth and is protected by United States copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE like any other copyrighted material (e.g., a book or musical recording) except that you may either (a) make one copy of the SOFTWARE solely for backup or archival purposes, or (b) transfer the SOFTWARE to a single hard disk provided you keep the original solely for backup or archival purposes. You may make a single copy of this document for your own use only.
3. OTHER RESTRICTIONS. You may not rent or lease the SOFTWARE, but you may transfer the software and accompanying documentation on a permanent basis provided you retain no copies and the recipient agrees to the terms of this Agreement. You may not reverse engineer, decompile, or disassemble the software. If the SOFTWARE is an update, any transfer must include the most recent update and all previous versions.

NO WARRANTIES. Cornel Huth disclaims all warranties, either expressed or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to the SOFTWARE and the documentation.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES. In no event shall Cornel Huth be liable for any damages whatsoever arising out of use of or inability to use this SOFTWARE.

U.S. GOVERNMENT RESTRICTED RIGHTS

The SOFTWARE and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013. Contractor/manufacturer is Cornel Huth/6402 Ingram Rd/San Antonio, TX 78238.(512)684-8065.

This agreement is governed by the laws of the state of Texas.

3

FUNCTION LIST

FUNCTION CreateFile(pathname\$,BYVAL attribute)
FUNCTION OpenDevice(pathname\$,BYVAL mode,handle,flen&)
FUNCTION ReadDevice(BYVAL han,BYVAL start&,BYVAL bytes&,BYVAL vseg,BYVAL voff)
FUNCTION WriteDevice(BYVAL han,BYVAL start&,BYVAL bytes&,BYVAL vseg,BYVAL voff)
FUNCTION CloseDevice(BYVAL handle)

CreateFile

| | |
|-----------|---|
| Type | FUNCTION - INTEGER |
| Arguments | pathname\$ - STRING, VAR-LEN attribute - INTEGER (BYVAL) |
| Syntax | stat = CreateFile(pathname\$,attribute) |
| Use | Create a new file using DOS function 3Ch. |
| Example 1 | pathname\$ = "C:\SND\MODINS.BNK" + CHR\$(0) attribute = 2 stat = CreateFile(pathname\$, attribute) IF stat THEN DoError stat |
| Rules | pathname\$ must not already exist. pathname\$ must be ASCIIIZ - end with a CHR\$(0). attribute values are: 0=normal, 1=read-only, 2=hidden, 4=system. |
| Notes | Critical errors such as open drive doors, unformatted disks, etc., are handled by QBXIOIOL. |
| Return | If stat = -1 then the pathname\$ either has zero length or is not ASCIIIZ. See DOS ERROR CODES below. |

OpenDevice

Type FUNCTION - INTEGER

Arguments pathname\$ - STRING, VAR-LEN
 mode - INTEGER (BYVAL)
 handle - INTEGER (returned)
 flen& - LONG (returned)

Syntax stat = OpenDevice(pathname\$,mode,handle,flen&)

Use Open a DOS file or device using DOS function 3Dh.

Example 1 pathname\$ = "C:\SND\MODINS.BNK" + CHR\$(0)
 mode = &H42 'full shared access, R/W open
 stat = OpenDevice(pathname\$, mode, handle, flen&)
 IF stat then DoError stat

Rules pathname\$ must exist. pathname\$ must be ASCIIIZ -
 end with a CHR\$(0). mode values are (bit-mapped):

 bit 7 = inheritance flag (DOS 3+)
 4-6 = sharing mode (DOS 3+)
 3 = reserved (should=0)

0-2 = access type

For read/write non-network access to a file use mode=2. For exact uses of mode consult a DOS programmer reference.

Notes Handle and file length (flen&) are valid only if stat is zero.

Critical errors such as open drive doors, unformatted disks, etc., are handled by QBXIOL.

Return If stat = -1 then the pathname\$ either has zero length or is not ASCIIIZ. See DOS ERROR CODES below.

ReadDevice

Type FUNCTION - INTEGER

Arguments handle - INTEGER (BYVAL)
start& - LONG (BYVAL)
bytes& - LONG (BYVAL)
vseg - INTEGER (BYVAL)
voff - INTEGER (BYVAL)

Syntax stat = ReadDevice(handle, start&, bytes&, vseg, voff)

Use Read from a DOS file or device to RAM using DOS function 3Fh.

Example 1 'open file (see above)
'allocate RAM buffer and point to it
REDIM buffer(0 TO flen&\8) AS STRING * 8

```
vseg = VARSEG(buffer(0))
voff = VARPTR(buffer(0))
'read 2016 bytes to buffer starting a byte
'32 (0-based, so the 33rd byte) in the file
stat = ReadDevice(handle,32&,2016&,vseg,voff)
IF stat THEN DoError stat
```

Rules First byte is byte 0.

Notes Don't read more bytes than space allocated to the buffer. To continue reading from the previous position use start& = -1.

Critical errors such as open drive doors, unformatted disks, etc., are handled by QBXIOL.

Return See DOS ERROR CODES below.

WriteDevice

Type FUNCTION - INTEGER

Arguments handle - INTEGER (BYVAL)
start& - LONG (BYVAL)
bytes& - LONG (BYVAL)
vseg - INTEGER (BYVAL)
voff - INTEGER (BYVAL)

Syntax stat = WriteDevice(handle,start&,bytes&,vseg,voff)

Use Write to a DOS file or device from RAM using DOS function 40h.

Example 1 'open file (see above)
'allocate RAM buffer and point to it
REDIM buffer(0 TO 32000) AS STRING * 8
vseg = VARSEG(buffer(0))
voff = VARPTR(buffer(0))
'fill buffer with data
FillBuffer buffer(), bytes&
'initial write to file
stat = WriteDevice(handle,0&,bytes&,vseg,voff)
IF stat THEN DoError stat

Rules First byte is byte 0.

Notes To continue writing to the previous position use start& = -1.

Critical errors such as open drive doors, unformatted disks, etc., are handled by QBXIOL.

You can input/output to the predefined DOS handles without needing to open them.

Return See DOS ERROR CODES below.

CloseDevice

Type FUNCTION - INTEGER

Arguments handle - INTEGER (BYVAL)

Syntax stat = CloseDevice(handle)

Use Close a DOS file using DOS function 3Eh.

Example 1 'open file (see above)
 'perform work (see above)
 'and when done close file
 stat = CloseDevice(handle)
 IF stat THEN DoError stat

Rules Handles 0-4 cannot be closed.

Notes Critical errors such as open drive doors, unfor-
 matted disks, etc., are handled by QBXIOL.

Return See DOS ERROR CODES below.

DOS ERROR CODES

Possible stat return codes in decimal.

- 2 disk full
- 1 bad filename
- 0 no error
- 1 function number invalid
- 2 file not found
- 3 path not found
- 4 too many open files
- 5 access denied
- 6 handle invalid
- 7 memory control blocks destroyed
- 8 insufficient memory
- 9 memory block address invalid
- 10 environment invalid
- 11 format invalid
- 12 access code invalid
- 13 data invalid
- 14 reserved
- 15 disk drive invalid
- 16 attempt to remove current directory
- 17 not same device
- 18 no more files