



DosQFSInfo

This call queries information from a file system device.

Syntax

```
DosQFSInfo (DriveNumber, FSInfoLevel, FSInfoBuf, FSInfoBufSize)
```

Parameters

:DriveNumber (USHORT) - input : Logical drive number (0 = default, 1 = A, and so on). :When a logical drive is specified, the media in the drive is examined (local drive only) and the request is passed to the FSD responsible for managing that media or to the FSD that is attached to the drive. ;FSInfoLevel (USHORT) - input : Level of file information required. ;FSInfoBuf (PBYTE) - output : Address of the storage area where the system returns the requested level of file information. :Level 1 Information :For FSInfoLevel = 1, information is returned in the following structure: ::filesysid (ULONG) - File system ID. ::sectornum (ULONG) - Number of sectors per allocation unit. ::unitnum (ULONG) - Number of allocation units. ::unitavail (ULONG) - Number of allocation units available. ::bytesnum (USHORT) - Number of bytes per sector. :Level 2 Information :For FSInfoLevel = 2, the information is returned in the following format: ::reserved (ULONG) - Reserved ::volumelength (BYTE) - Length of the volume label, not including the null. ::volumelabel (CHAR) - Volume label ASCIIIZ string. ;FSInfoBufSize (USHORT) : Length of the buffer.

Return Code

rc (USHORT) - return Return code descriptions are: * 0 NO_ERROR * 15 ERROR_INVALID_DRIVE * 111 ERROR_BUFFER_OVERFLOW * 124 ERROR_INVALID_LEVEL * 125 ERROR_NO_VOLUME_LABEL

Remarks

Trailing blanks supplied at volume label definition time are not considered to be part of the label and are therefore not returned as valid label data. Volume label is limited to a length of 11 bytes.

Volume Serial Number is a unique 32-bit number used by OS/2 to positively identify its disk/diskette volumes. The hard error prompts the user for an unmounted removable volume by displaying both the Volume Serial Number (as an 8 digit hexadecimal number) and the Volume Label.

If there is no volume serial number on the disk/diskette, the volume serial number information is returned as binary zeros. If there is no volume label on the disk/diskette, the volume label information is returned as blank spaces. If there is no volume serial number and/or volume label for disk/diskette volumes formatted by DOS 3.X, this information is not displayed by the Hard Error Handler.

Example Code**C Binding**

```
<PRE> typedef struct _FSALLOCATE {

    ULONG idFileSystem; /* file system ID */
    ULONG cSectorUnit; /* number sectors per allocation unit */
    ULONG cUnit; /* number of allocation units */
    ULONG cUnitAvail; /* available allocation units */
    USHORT cbSector; /* bytes per sector */

} FSALLOCATE;

typedef struct _FDATE { /* fdate */

    unsigned day : 5; /* binary day for directory entry */

    unsigned month : 4; /* binary month for directory entry */
    unsigned year : 7; /* binary year for directory entry */

} FDATE;

typedef struct _FTIME { /* ftime */

    unsigned twosecs : 5; /* binary number of two-second increments */

    unsigned minutes : 6; /* binary number of minutes */
    unsigned hours : 5; /* binary number of hours */

} FTIME;

typedef struct _FSINFO { /* fsinf */

    FDATE fdateCreation;
    FTIME ftimeCreation;
    VOLUMELABEL vol;

} FSINFO;

typedef struct _VOLUMELABEL { /* vol */

    BYTE cch;
    CHAR szVolLabel[12];

} VOLUMELABEL;

#define INCL_DOSFILEMGR

USHORT rc = DosQFSInfo(DriveNumber, FSInfoLevel, FSInfoBuf, FSInfoBufSize);
```

```
USHORT DriveNumber; /* Drive number */ USHORT FSInfoLevel; /* File system data required */ PBYTE  
FSInfoBuf; /* File system info buffer */ USHORT FSInfoBufSize; /* File system info buffer size */  
  
USHORT rc; /* return code */ </PRE>
```

MASM Binding

```
<PRE> FSALLOCATE struc
```

fsalloc_idFileSystem	dd	?
fsalloc_cSectorUnit	dd	?
fsalloc_cUnit	dd	?
fsalloc_cUnitAvail	dd	?
fsalloc_cbSector	dw	?

```
FSALLOCATE ends
```

```
EXTRN DosQFSInfo:FAR INCL_DOSFILEMGR EQU 1
```

```
PUSH WORD DriveNumber ;Drive number PUSH WORD FSInfoLevel ;File system data required PUSH@  
OTHER FSInfoBuf ;File system info buffer (returned) PUSH WORD FSInfoBufSize ;File system info buffer  
size CALL DosQFSInfo
```

Returns WORD

```
</PRE>
```

Note

Text based on <http://www.edm2.com/index.php/DosQFSInfo>

Family API	
DOS	Process Manager DosBeep DosExit DosSleep DosExecPgm
	File Manager DosChDir DosChgFilePtr DosClose DosDelete DosDupHandle DosMkDir DosMove DosQCurDir DosQCurDisk DosSet FileMode DosOpen DosQFileInfo DosRead DosQ FileMode DosQFSInfo DosQVerify DosRmDir DosSelectDisk DosFindClose DosFindFirst DosFindNext DosSet FileInfo DosSet Verify DosWrite DosFileLocks DosSet FHandState DosNewSize DosBufReset DosQFHandState DosSet FInfo
	Memory Manager DosFreeSeg DosSubAlloc DosSubFree DosSubSet DosAlloc Huge DosAlloc Seg DosRealloc Huge DosRealloc Seg DosGet Huge Shift DosCreate CS Alias
	NLS DosCaseMap DosGet Ctry Info DosGet DBCSEv DosSet Ctry Code DosGet Collate DosGet Message DosIns Message DosPut Message
	Date and Time DosSet Date Time DosGet Date Time
	Devices DosDevConfig DosDevIOCtl DosDevIOCtl2
	Signals DosHoldSignal DosSet Sig Handler
	Misc BadDynLink DosGet Env DosGet Machine Mode DosGet Version DosError DosErr Class DosSet Vec
KBD	KbdCharIn KbdFlushBuffer KbdGet Status KbdSet Status KbdStringIn KbdPeek
VIO	VioGet Buf VioGet Config VioGet Cur Pos VioGet Cur Type VioGet Phys Buf VioRead Cell Str VioRead Char Str VioScroll Up VioScroll Dn VioScroll If VioScroll Rt VioScr Un Lock VioSet Cur Pos VioSet Cur Type VioSet Mode VioGet Mode VioShow Buf VioWrt Cell Str VioWrt Char Str VioWrt Char Str Att VioWrt N Attr VioWrt N Cell VioWrt N Char VioWrt TTY VioScr Lock VioPop Up
Tools	BIND
Modules	DOSCALLS.DLL VIOCALS.DLL KBDCALS.DLL MSG.DLL
Libraries	API.LIB OS2386.LIB FAPI.LIB DOSCALLS.LIB SUBCALLS.LIB

2018/08/25 15:05 · prokushev · 0 Comments

From:
<https://ftp.osfree.org/doku/> - osFree wiki



Permanent link:
<https://ftp.osfree.org/doku/doku.php?id=en:docs:fapi:dosqfsinfo&rev=1607086656>

Last update: **2020/12/04 12:57**