

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 ASCII 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	<a href="#">DosBeep</a> <a href="#">DosExit</a> <a href="#">DosSleep</a> <a href="#">DosExecPgm</a>
	File Manager	<a href="#">DosChDir</a> <a href="#">DosChgFilePtr</a> <a href="#">DosClose</a> <a href="#">DosDelete</a> <a href="#">DosDupHandle</a> <a href="#">DosMkDir</a> <a href="#">DosMove</a> <a href="#">DosQCurDir</a> <a href="#">DosQCurDisk</a> <a href="#">DosSetFileMode</a> <a href="#">DosOpen</a> <a href="#">DosQFileInfo</a> <a href="#">DosRead</a> <a href="#">DosQFileMode</a> <a href="#">DosQFSInfo</a> <a href="#">DosQVerify</a> <a href="#">DosRmDir</a> <a href="#">DosSelectDisk</a> <a href="#">DosFindClose</a> <a href="#">DosFindFirst</a> <a href="#">DosFindNext</a> <a href="#">DosSetFileInfo</a> <a href="#">DosSetVerify</a> <a href="#">DosWrite</a> <a href="#">DosFileLocks</a> <a href="#">DosSetFHandState</a> <a href="#">DosNewSize</a> <a href="#">DosBufReset</a> <a href="#">DosQFHandState</a> <a href="#">DosSetFSInfo</a>
	Memory Manager	<a href="#">DosFreeSeg</a> <a href="#">DosSubAlloc</a> <a href="#">DosSubFree</a> <a href="#">DosSubSet</a> <a href="#">DosAllocHuge</a> <a href="#">DosAllocSeg</a> <a href="#">DosReallocHuge</a> <a href="#">DosReallocSeg</a> <a href="#">DosGetHugeShift</a> <a href="#">DosCreateCSAlias</a>
	NLS	<a href="#">DosCaseMap</a> <a href="#">DosGetCtryInfo</a> <a href="#">DosGetDBCSEv</a> <a href="#">DosSetCtryCode</a> <a href="#">DosGetCollate</a> <a href="#">DosGetMessage</a> <a href="#">DosInsMessage</a> <a href="#">DosPutMessage</a>
	Date and Time	<a href="#">DosSetDateTime</a> <a href="#">DosGetDateTime</a>
	Devices	<a href="#">DosDevConfig</a> <a href="#">DosDevIOCtl</a> <a href="#">DosDevIOCtl2</a>
	Signals	<a href="#">DosHoldSignal</a> <a href="#">DosSetSigHandler</a>
	Misc	<a href="#">BadDynLink</a> <a href="#">DosGetEnv</a> <a href="#">DosGetMachineMode</a> <a href="#">DosGetVersion</a> <a href="#">DosError</a> <a href="#">DosErrClass</a> <a href="#">DosSetVec</a>
KBD		<a href="#">KbdCharIn</a> <a href="#">KbdFlushBuffer</a> <a href="#">KbdGetStatus</a> <a href="#">KbdSetStatus</a> <a href="#">KbdStringIn</a> <a href="#">KbdPeek</a>

Family API	
VIO	<a href="#">VioGetBuf</a> <a href="#">VioGetConfig</a> <a href="#">VioGetCurPos</a> <a href="#">VioGetCurType</a> <a href="#">VioGetPhysBuf</a> <a href="#">VioReadCellStr</a> <a href="#">VioReadCharStr</a> <a href="#">VioScrollUp</a> <a href="#">VioScrollDn</a> <a href="#">VioScrollLf</a> <a href="#">VioScrollRt</a> <a href="#">VioScrUnLock</a> <a href="#">VioSetCurPos</a> <a href="#">VioSetCurType</a> <a href="#">VioSetMode</a> <a href="#">VioGetMode</a> <a href="#">VioShowBuf</a> <a href="#">VioWrtCellStr</a> <a href="#">VioWrtCharStr</a> <a href="#">VioWrtCharStrAtt</a> <a href="#">VioWrtNAttr</a> <a href="#">VioWrtNCell</a> <a href="#">VioWrtNChar</a> <a href="#">VioWrtTTY</a> <a href="#">VioScrLock</a> <a href="#">VioPopUp</a>
Tools	<a href="#">BIND</a>
Modules	<a href="#">DOSCALLS.DLL</a> <a href="#">VIOCALLS.DLL</a> <a href="#">KBDCALLS.DLL</a> <a href="#">MSG.DLL</a>
Libraries	<a href="#">API.LIB</a> <a href="#">OS2386.LIB</a> <a href="#">FAPI.LIB</a> <a href="#">DOSCALLS.LIB</a> <a href="#">SUBCALLS.LIB</a>

2018/08/25 15:05 · [prokushev](#) · [0 Comments](#)

From:

<http://www.osfree.org/doku/> - **osFree wiki**

Permanent link:

<http://www.osfree.org/doku/doku.php?id=en:docs:fapi:dosqfsinfo&rev=1535543890>

Last update:

**2018/08/29 11:58**

