DosQFileInfo

This call returns information for a specific file.

Syntax

DosQFileInfo (FileHandle, FileInfoLevel, FileInfoBuf, FileInfoBufSize)

Parameters

;FileHandle (HFILE) - input : File handle. ;FileInfoLevel (USHORT) - input : Level of file information required. A value of 1, 2, or 3 can be specified. Level 4 is reserved. The structures described in FileInfoBuf indicate the information returned for each of these levels. ;FileInfoBuf (PBYTE) - output : Address of the storage area where the system returns the requested level of file information. File information, where applicable, is at least as accurate as the most recent DosClose, DosBufReset, DosSetFileInfo, or DosSetPathInfo.

'Level 1 Information'

FileInfoBuf contains the following structure, to which file information is returned:

filedate (FDATE) : Structure containing the date of file creation. 'Bit Description' 15-9 Year, in binary, of file creation 8-5 Month, in binary, of file creation 4-0 Day, in binary, of file creation.

filetime (FTIME) : Structure containing the time of file creation.

'Bit Description' 15-11 Hours, in binary, of file creation 10-5 Minutes, in binary, of file creation 4-0 Seconds, in binary number of two-second increments, of file creation.

fileaccessdate (FDATE) : Structure containing the date of last access. See FDATE in filedate.

fileaccesstime (FTIME) : Structure containing the time of last access. See FTIME in filetime.

writeaccessdate (FDATE) : Structure containing the date of last write. See FDATE in filedate.

writeaccesstime (FTIME) : Structure containing the time of last write. See FTIME in filetime.

filesize (ULONG) : File size.

filealloc (ULONG) : Allocated file size.

fileattrib (USHORT) : Attributes of the file, defined in DosSetFileMode.

'Level 2 Information'

FileInfoBuf contains a structure similar to the Level 1 structure, with the addition of the cbList field after the fileattrib field.

cbList (ULONG) : On output, this field contains the length of the entire EA set for the file object. This value can be used to calculate the size of the buffer required to hold EA information returned when FileInfoLevel = 3 is specified.

'Level 3 Information'

FileInfoBuf contains an EAOP structure, which has the following format:

fpGEAList (PGEALIST) : Address of GEAList. GEAList is a packed array of variable length “get EA” structures, each containing an EA name and the length of the name.

fpFEAList (PFEALIST) : Address of FEAList. FEAList is a packed array of variable length “full EA” structures, each containing an EA name and its corresponding value, as well as the lengths of the name and the value.

oError (ULONG) : Offset into structure where error has occurred.

On input, FileInfoBuf is an EAOP structure. fpGEAList points to a GEA list defining the attribute names whose values are returned. fpFEAList points to a data area where the relevant FEA list is returned. The length field of this FEA list is valid, giving the size of the FEA list buffer. oError points to the offending GEA entry in case of error. Following is the format of the GEAList structure:

cbList (ULONG) : Length of the GEA list, including the length itself.

list (GEA) : List of GEA structures. A GEA structure has the following format:

cbName (BYTE) : Length of EA ASCIIZ name, which does not include the null character.

szName (CHAR) : ASCIIZ name of EA.

On output, FileInfoBuf is unchanged. The buffer pointed to by fpFEAList is filled in with the returned information. If the buffer fpFEAList points to isn't large enough to hold the returned information (ERROR_BUFFER_OVERFLOW) cbList is still valid, assuming there's at least enough space for it. Its value is the size of the entire EA set for the file, even though only a subset of attributes was requested. Following is the format of the FEAList structure:

cbList (ULONG) : Length of the FEA list, including the length itself.

list (FEA) : List of FEA structures. An FEA structure has the following format:

Flags (BYTE) : Bit indicator describing the characteristics of the EA being defined.

'Bit Description' 7 Critical EA. 6-0 Reserved and must be set to zero. If bit 7 is set to 1, this indicates a critical EA. If bit 7 is 0, this means the EA is noncritical; that is, the EA is not essential to the intended use by an application of the file with which it is associated.

cbName (BYTE) : Length of EA ASCIIZ name, which does not include the null character.

cbValue (USHORT) : Length of EA value, which cannot exceed 64KB.

szName (PSZ) : Address of the ASCIIZ name of EA.

aValue (PSZ) : Address of the free-format value of EA.

'Note:' The szName and aValue fields are not included as part of header or include files. Because of their variable lengths, these entries must be built manually.

; FileInfoBufSize (USHORT) - output : Length of FileInfoBuf.

Return Code

rc (USHORT) - return

Return code descriptions are:

* 0 NO_ERROR * 5 ERROR_ACCESS_DENIED * 6 ERROR_INVALID_HANDLE * 111 ERROR_BUFFER_OVERFLOW * 124 ERROR_INVALID_LEVEL * 130 ERROR_DIRECT_ACCESS_HANDLE * 254 ERROR_INVALID_EA_NAME * 255 ERROR_EA_LIST_INCONSISTENT

Remarks

The FAT file system supports modification of only date and time information contained in file information level 1. Zero is returned for the creation and access dates and times.

To return information contained in any of the file information levels, DosQFileInfo has to be able to read the open file. DosQFileInfo works only when the file is opened for read access, with a deny-write sharing mode specified for access by other processes. If the file is already opened by another process that has specified conflicting sharing and access modes, a call to DosOpen or DosOpen2 fails.

DosEnumAttribute returns the lengths of EAs. This information can be used to calculate the size of FileInfoBuf needed to hold full extended attribute (FEA) information returned by DosQFileInfo when Level 3 is specified. The size of the buffer is calculated as follows:

* One byte (for fea_usFlags) + * One byte (for fea_cbName) + * Two bytes (for fea_cbValue) + * Value of cbName (for the name of the EA) + * One byte (for terminating NULL in fea_cbName) + * Value of cbValue (for the value of the EA)

Example Code

C Binding

<PRE> 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 _FILESTATUS { /* fsts */

FDATE fdateCreation; /* date of file creation */

```
FTIME  ftimeCreation;        /* time of file creation */
FDATE  fdateLastAccess;      /* date of last access */
FTIME  ftimeLastAccess;      /* time of last access */
FDATE  fdateLastWrite;       /* date of last write */
FTIME  ftimeLastWrite;       /* time of last write */
ULONG  cbFile;               /* file size (end of data) */
ULONG  cbFileAlloc;          /* file allocated size */
USHORT attrFile;             /* attributes of the file */
```

} FILESTATUS;

typedef struct _GEA { /* gea */

BYTE cbName; /* name length not including NULL */

```
CHAR szName[1];          /* attribute name */
```

} GEA;

typedef struct _GEALIST { /* geal */

ULONG cbList; /* total bytes of structure including full list */

```
GEA list[1];             /* variable length GEA structures */
```

} GEALIST;

typedef struct _FEA { /* fea */

BYTE fEA; /* flags */

```
BYTE cbName;             /* name length not including NULL */
USHORT cbValue;          /* value length */
```

} FEA;

typedef struct _FEALIST { /* feal */

ULONG cbList; /* total bytes of structure including full list */

```
FEA list[1];            /* variable length FEA structures */
```

} FEALIST;

typedef struct _EAOP { /* eaop */

PGEALIST fpGEAList; /* general EA list */

```
PFEALIST fpFEAList;     /* full EA list */
ULONG  oError;
```

} EAOP;

#define INCL_DOSFILEMGR

USHORT rc = DosQFileInfo(FileHandle, FileInfoLevel, FileInfoBuf,

```
                              FileInfoBufSize);
```

HFILE FileHandle; /* File handle */ USHORT FileInfoLevel; /* File data required */ PBYTE FileInfoBuf; /* File data buffer (returned) */ USHORT FileInfoBufSize; /* File data buffer size */

USHORT rc; /* return code */

</PRE>

MASM Binding

<PRE> FDATE struc

fdate_fs dw ?

FDATE ends

FTIME struc

ftime_fs dw ?

FTIME ends

FILESTATUS struc

fsts_fdateCreation dw (size FDATE)/2 dup (?) ;date of file creation

```
fsts_ftimeCreation   dw (size FTIME)/2 dup (?) ;time of file creation
fsts_fdateLastAccess dw (size FDATE)/2 dup (?) ;date of last access
fsts_ftimeLastAccess dw (size FTIME)/2 dup (?) ;time of last access
```

```
fsts_fdateLastWrite  dw (size FDATE)/2 dup (?) ;date of last write
fsts_ftimeLastWrite  dw (size FTIME)/2 dup (?) ;time of last write
fsts_cbFile          dd  ? ;file size (end of data)
fsts_cbFileAlloc     dd  ? ;file allocated size
fsts_attrFile        dw  ? ;attributes of the file
```

FILESTATUS ends

GEA struc

gea_cbName db ? ;name length not including NULL

```
gea_szName      db  1 dup (?)  ;attribute name
```

GEA ends

GEALIST struc

geal_cbList dd ? ;total bytes of structure including full list

```
geal_list       db  size GEA * 1 dup (?) ;variable length GEA structures
```

GEALIST ends

FEA struc

fea_fEA db ? ;flags

```
fea_cbName      db  ? ;name length not including NULL
fea_cbValue     dw  ? ;value length
```

FEA ends

FEALIST struc

feal_cbList dd ? ;total bytes of structure including full list

```
feal_list       db  size FEA * 1 dup (?) ;variable length FEA structures
```

FEALIST ends

EAOP struc

eaop_fpGEAList dd ? ;general EA list

```
eaop_fpFEAList  dd  ? ;full EA list
eaop_oError     dd  ? ;
```

EAOP ends

EXTRN DosQFileInfo:FAR INCL_DOSFILEMGR EQU 1

PUSH WORD FileHandle ;File handle PUSH WORD FileInfoLevel ;File data required PUSH@ OTHER FileInfoBuf ;File data buffer (returned) PUSH WORD FileInfoBufSize ;File data buffer size CALL DosQFileInfo

Returns WORD </PRE>

Note

Text based on http://www.edm2.com/index.php/DosQFileInfo

Family API		
DOS	Process Manager	DosBeep DosExit DosSleep DosExecPgm
	File Manager	DosChDir DosChgFilePtr DosClose DosDelete DosDupHandle DosMkDir DosMove DosQCurDir DosQCurDisk DosSetFileMode DosOpen DosQFileInfo DosRead DosQFileMode DosQFSInfo DosQVerify DosRmDir DosSelectDisk DosFindClose DosFindFirst DosFindNext DosSetFileInfo DosSetVerify DosWrite DosFileLocks DosSetFHandState DosNewSize DosBufReset DosQFHandState DosSetFSinfo DosShutdown
	Memory Manager	DosFreeSeg DosSubAlloc DosSubFree DosSubSet DosAllocHuge DosAllocSeg DosReallocHuge DosReallocSeg DosGetHugeShift DosCreateCSAlias
	NLS	DosCaseMap DosGetCtryInfo DosGetDBCSEv DosSetCtryCode DosGetCollate DosGetMessage DosInsMessage DosPutMessage
	Date and Time	DosSetDateTime DosGetDateTime
	Devices	DosDevConfig DosDevIOCtl DosDevIOCtl2
	Signals	DosHoldSignal DosSetSigHandler
	Misc	BadDynLink DosGetEnv DosGetMachineMode DosGetVersion DosError DosErrClass DosSetVec
KBD		KbdCharIn KbdFlushBuffer KbdGetStatus KbdSetStatus KbdStringIn KbdPeek
VIO		VioGetBuf VioGetConfig VioGetCurPos VioGetCurType VioGetPhysBuf VioReadCellStr VioReadCharStr VioScrollUp VioScrollDn VioScrollLf VioScrollRt VioScrUnLock VioSetCurPos VioSetCurType VioSetMode VioGetMode VioShowBuf VioWrtCellStr VioWrtCharStr VioWrtCharStrAtt VioWrtNAttr VioWrtNCell VioWrtNChar VioWrtTTY VioScrLock VioPopUp
Tools		BIND
Modules		DOSCALLS.DLL VIOCALLS.DLL KBDCALLS.DLL MSG.DLL
Libraries		API.LIB OS2386.LIB FAPI.LIB DOSCALLS.LIB SUBCALLS.LIB

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