



This is part of **Family API** which allow to create dual-os version of program runs under OS/2 and DOS

Note: This is legacy API call. It is recommended to use 32-bit equivalent

2021/09/17 04:47 · prokushev · [0 Comments](#)

2021/08/20 03:18 · prokushev · [0 Comments](#)

DosSetFileInfo

This call sets attribute and extended attribute information for a file.

Syntax

```
DosSetFileInfo (FileHandle, FileInfoLevel, FileInfoBuf, FileInfoBufSize)
```

Parameters

- FileHandle ([HFILE](#)) - input : File handle.
- FileInfoLevel ([USHORT](#)) - input : Level of file information being set. A value of 1 or 2 can be specified. The structures described in FileInfoBuf indicate the information being set for each of these levels.
- FileInfoBuf ([PBYTE](#)) - input : Address of the storage area containing the structures for file information levels.

; FileInfoBufSize ([USHORT](#)) - input : Length of FileInfoBuf.

Level 1 Information

FileInfoBuf contains the following structure, to which information is specified in the following format:

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

Bit	Description
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 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.

Level 2 sets a series of EA name/value pairs. On input, FileInfoBuf is an EAOP structure above. fpGEAList is ignored. fpFEAList points to a data area where the relevant FEA list is to be found. oError is ignored. 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.

On output, fpGEAList is unchanged. fpFEAList is unchanged as is the area pointed to by fpFEAList. If an error occurred during the set, oError is the offset of the FEA where the error occurred. The API return code is the error code corresponding to the condition generating the error. If no error occurred, oError is undefined.

Return Code

- 0 NO_ERROR
- 1 ERROR_INVALID_FUNCTION
- 5 ERROR_ACCESS_DENIED
- 6 ERROR_INVALID_HANDLE
- 87 ERROR_INVALID_PARAMETER
- 122 ERROR_INSUFFICIENT_BUFFER
- 124 ERROR_INVALID_LEVEL
- 130 ERROR_DIRECT_ACCESS_HANDLE
- 254 ERROR_INVALID_EA_NAME
- 255 ERROR_EA_LIST_INCONSISTENT

Remarks

DosSetFileInfo is successful only when the file is opened for write access, with a deny-both sharing mode specified for access to the file by other processes. If the file is already opened with conflicting sharing rights, the call to [DosOpen](#) or [DosOpen2](#) will fail.

A 0 value in the date and time components of a field does not change the field. For example, if both "last write date" and "last write time" are specified as 0 in the Level 1 information structure, then both attributes of the file are left unchanged. If either "last write date" or "last write time" are specified as non-zero, both attributes of the file are set to the new values.

The FAT file system supports modification of only the dates and times of the last write. Creation and last access dates and times are not affected.

The last modification date and time will get changed if the extended attributes are modified.

Family API Considerations

It is not possible to create a label with leading blank characters in DOS mode, because of restrictions on the previous Interrupt 21h function call (create an FCB type file), which must be used by FAPI.

Example Code

C Binding

```
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 = DosSetFileInfo(FileHandle, FileInfoLevel, FileInfoBuf,
                           FileInfoBufSize);

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

USHORT     rc; /* return code */

```

MASM Binding

```

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 struct

```
gea_cbName    db ? ;name length not including NULL
gea_szName    db 1 dup (?) ;attribute name
```

GEA ends

GEALIST struct

```
geal_cbList    dd ? ;total bytes of structure including full list
geal_list      db size GEA * 1 dup (?) ;variable length GEA structures
```

GEALIST ends

FEA struct

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

FEA ends

FEALIST struct

```
feal_cbList    dd ? ;total bytes of structure including full list
feal_list      db size FEA * 1 dup (?) ;variable length FEA structures
```

FEALIST ends

EAOP struct

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

EAOP ends

EXTRN DosSetFileInfo:FAR
 INCL_DOSFILEMGR EQU 1

```

PUSH WORD FileHandle ;File handle
PUSH WORD FileInfoLevel ;File info data required
PUSH@ OTHER FileInfoBuf ;File info buffer
PUSH WORD FileInfoBufSize ;File info buffer size
CALL DosSetFileInfo

```

Returns WORD

Note

Text based on [http://www.edm2.com/index.php/DosSetFileInfo_\(FAP\)](http://www.edm2.com/index.php/DosSetFileInfo_(FAP))

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 DosDevIOct1 DosDevIOct2
	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

From: <https://osfree.ru/doku/> - **osFree wiki**

Permanent link: <https://osfree.ru/doku/doku.php?id=en:docs:fapi:dossetfileinfo&rev=1631867493>

Last update: **2021/09/17 08:31**



