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

DosQPathInfo returns attribute and extended attribute information for a file or subdirectory.

Syntax

DosQPathInfo (PathName, PathInfoLevel, PathInfoBuf, PathInfoBufSize, Reserved)

Parameters

;PathName (PSZ) - input: Address of the ASCIIZ full path name of the file or subdirectory. Global file name characters can be used in the name only for PathInfoLevels five and six. : DosQSysInfo is called by an application during initialization to determine the maximum path length allowed by OS/2. ;PathInfoLevel (USHORT) - input: Level of path information required. A value of 1, 2, 3, 5, or 6 can be specified. Level 4 is reserved. The structures described in PathInfoBuf indicate the information returned for each of these levels.; PathInfoBuf (PBYTE) - output: Address of the storage area containing the requested level of path information. Path information, where applicable, is based on the most recent DosClose, DosBufReset, DosSetFileInfo, or DosSetPathInfo. :;Level 1 Information :PathInfoBuf contains the following structure, to which path information is returned: ::filedate (FDATE) - Structure containing the date of creation. :::15-9 - Year, in binary, of creation :::8-5 - Month, in binary, of creation :::4-0 - Day, in binary, of creation. ::filetime (FTIME) - Structure containing the time of creation:::15-11 - Hours, in binary, of creation:::10-5 - Minutes, in binary, of creation:::4-0 -Seconds, in binary number of two-second increments, of 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: PathInfoBuf 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 PathInfoLevel = 3 is specified. :;Level 3 Information :PathInfoBuf 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, PathInfoBuf 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

Last update: 2021/11/04 13:56

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, PathInfoBuf 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. ::: 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. :;Level 5 Information ::Level 5 returns the fully qualified ASCIIZ name of PathName in PathInfoBuf. The PathName may contain global file name characters. :;Level 6 Information ::Level 6 requests a file system to verify the correctness of PathName per its rules of syntax. An erroneous name is indicated by an error return code. The PathName may contain global file name characters. ;PathInfoBufSize (USHORT) - output: Length of PathInfoBuf.; Reserved (ULONG) - input: Reserved, must be set to zero.

Return Code

;rc (USHORT) - return:Return code descriptions are: * 0 NO_ERROR * 3 ERROR_PATH_NOT_FOUND *32 ERROR SHARING VIOLATION *111 ERROR BUFFER OVERFLOW *124 ERROR INVALID LEVEL *206 ERROR FILENAME EXCED RANGE *254 ERROR INVALID EA NAME *255 ERROR EA LIST INCONSISTENT

Remarks

For DosQPathInfo to return information contained in any of the file information levels, the file object must be opened for read access, with a deny-write sharing mode specified for access by other processes. Thus, if the file object is already accessed by another process that holds conflicting sharing and access rights, a call to DosQPathInfo fails.

Bindings

C

<PRE> typedef struct FDATE { /* fdate */

```
/* binary day for directory entry */
unsigned day
               : 5;
                        /* binary month for directory entry */
unsigned month: 4;
unsigned year
                        /* binary year for directory entry */
               : 7;
```

2025/11/03 09:22 3/6 DosQPathInfo

```
} FDATE;
typedef struct FTIME { /* ftime */
unsigned twosecs : 5;
                              /* binary number of two-second increments */
                              /* binary number of minutes */
unsigned minutes : 6;
                              /* binary number of hours */
unsigned hours
                : 5;
} 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 */
                               /* date of last write */
FDATE fdateLastWrite;
                               /* time of last write */
FTIME ftimeLastWrite;
                               /* file size (end of data) */
ULONG cbFile:
                               /* file allocated size */
ULONG cbFileAlloc;
                               /* attributes of the file */
USHORT attrFile;
} FILESTATUS;
typedef struct GEA { /* gea */
                          /* name length not including NULL */
BYTE cbName;
                          /* attribute name */
CHAR szName[1];
} GEA;
typedef struct _GEALIST { /* geal */
                          /* total bytes of structure including full list */
ULONG cbList:
                          /* variable length GEA structures */
GEA list[1];
} 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 */
```

Last update: 2021/11/04

```
} FEALIST;
```

typedef struct EAOP { /* eaop */

```
PGEALIST fpGEAList;
                        /* general EA list */
PFEALIST fpFEAList;
                        /* full EA list */
ULONG oError;
```

} EAOP;

#define INCL DOSFILEMGR

USHORT rc = DosQPathInfo(PathName, PathInfoLevel, PathInfoBuf,

```
PathInfoBufSize, 0);
```

PSZ PathName; /* File or directory path name string */ USHORT PathInfoLevel; /* Data required */ PBYTE PathInfoBuf; /* Data buffer (returned) */ USHORT PathInfoBufSize; /* Data buffer size */ ULONG 0; /* Reserved (must be zero) */

USHORT rc; /* return code */ </PRE>

MASM

<PRE> FDATE struc

```
fdate fs
              ?
          dw
```

FDATE ends

FTIME struc

```
ftime fs
               ?
          dw
```

FTIME ends

FILESTATUS struc

```
dw (size FDATE)/2 dup (?) ;date of file creation
fsts fdateCreation
                     dw (size FTIME)/2 dup (?) ;time of file creation
fsts ftimeCreation
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
                        ? ;file size (end of data)
                     dd
fsts cbFileAlloc
                     dd ? ;file allocated size
fsts_attrFile
                       ? ;attributes of the file
```

FILESTATUS ends

2025/11/03 09:22 5/6 DosQPathInfo

GEA struc

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

GEA ends

GEALIST struc

GEALIST ends

FEA struc

FEA ends

FEALIST struc

FEALIST ends

EAOP struc

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

EAOP ends

EXTRN DosQPathInfo:FAR INCL DOSFILEMGR EQU 1

PUSH@ ASCIIZ PathName ;File or directory path name string PUSH WORD PathInfoLevel ;Data required PUSH@ OTHER PathInfoBuf ;Data buffer (returned) PUSH WORD PathInfoBufSize ;Data buffer size PUSH DWORD 0 ;Reserved (must be zero) CALL DosQPathInfo

Returns WORD </PRE>

Dos16

From:

http://185.82.219.184/doku/ - osFree wiki

Permanent link:

Last update: 2021/11/04 13:56

