2025/11/12 14:16 1/5 DosQPathInfo

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 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 */
```

```
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;

https://osfree.org/doku/ Printed on 2025/11/12 14:16

2025/11/12 14:16 3/5 DosQPathInfo

```
typedef struct FILESTATUS { /* fsts */
```

```
/* date of file creation */
FDATE fdateCreation;
                               /* time of file creation */
FTIME ftimeCreation:
FDATE fdateLastAccess;
                               /* date of last access */
FTIME ftimeLastAccess;
                               /* time of last access */
FDATE fdateLastWrite;
                               /* date of last write */
                               /* time of last write */
FTIME ftimeLastWrite;
                               /* file size (end of data) */
ULONG cbFile;
ULONG cbFileAlloc;
                               /* file allocated size */
                               /* 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;
GEA list[1];
                          /* variable length GEA structures */
} GEALIST;
typedef struct FEA { /* fea */
BYTE fEA:
                          /* flags */
                          /* name length not including NULL */
BYTE cbName;
                          /* value length */
USHORT cbValue;
} FEA;
typedef struct FEALIST { /* feal */
                          /* total bytes of structure including full list */
ULONG cbList;
                          /* variable length FEA structures */
FEA list[1];
} FEALIST;
typedef struct _EAOP { /* eaop */
PGEALIST fpGEAList;
                         /* general EA list */
                         /* full EA list */
PFEALIST fpFEAList;
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

```
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
                        ? ;file size (end of data)
                     dd
fsts cbFileAlloc
                        ? ;file allocated size
                     dd
fsts attrFile
                     dw
                        ? ;attributes of the file
```

FILESTATUS ends

**GEA struc** 

**GEA** ends

**GEALIST struc** 

https://osfree.org/doku/ Printed on 2025/11/12 14:16

| <pre>geal_list</pre> | db | size GEA * | 1 dup | (?) | :variable | lenath | GEA structures |
|----------------------|----|------------|-------|-----|-----------|--------|----------------|
| 900.1_1_0            |    | ·          |       | ,   | ,         |        | 0 = 1          |

# **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:

https://osfree.org/doku/ - osFree wiki

Permanent link:

https://osfree.org/doku/doku.php?id=en:docs:fapi:dosqpathinfo&rev=1632030629

Last update: **2021/09/19 05:50** 

