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

DosFindFirst2

This call finds the first file object or group of file objects whose name(s) match the specification. The specification can include extended attribute information associated with a file or subdirectory.

Syntax

```
DosFindFirst2 (FileName, DirHandle, Attribute, ResultBuf,
               ResultBufLen, SearchCount, FileInfoLevel, Reserved)
```

Parameters

- FileName (PSZ) - input : Address of the ASCIIZ path name of the file or subdirectory to be found. The name component may contain global file name characters.
- DirHandle (PHDIR) - input/output : Address of the handle associated with a specific DosFindFirst2 request. The values that can be specified are:
  - 0001H - The system assigns the handle for standard output, which is always available to a process.
  - FFFFH - The system allocates and returns a handle. If on input FFFFH is specified, on output DirHandle contains the handle allocated by the system.

The DosFindFirst2 handle is used with subsequent DosFindNext requests. Reuse of this handle in another DosFindFirst2 closes the association with the previous DosFindFirst2 and opens a new association.

- Attribute (USHORT) - input: Attribute value that determines the file objects to be searched for.

15-6	Reserved and must be zero
5	File archive
4	Subdirectory
3	Reserved and must be zero
2	System file
1	Hidden file
0	Read only file

These bits may be set individually or in combination. For example, an attribute value of 0021H (bits 5

and 0 set to 1) indicates a read-only file that should be archived.

To look at all directory entries except the volume label, set Attribute to hidden+system+directory (all three bits on). Attribute cannot specify the volume label. Volume labels are queried using DosQFSInfo.

If Attribute is 0, only normal file entries are found. Entries for subdirectories, hidden, and system files, are not returned.

- ResultBuf (PVOID) - input/output: Address of the directory search structures for file object information levels 1 through 3. The structure required for ResultBuf is dependent on the value specified for FileInfoLevel. The information returned reflects the last DosSetFileInfo, DosSetPathInfo, DosClose, and DosBufReset calls.

Level 1 Information

ResultBuf contains the following structure, to which directory entry information is returned:

- filedate (FDATE) : Structure containing the date of file creation.

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.

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.
- length (UCHAR) : Length of the ASCIIZ name string.
- matchfilename (CHAR) : ASCIIZ name string for the first occurrence of FileName.

Level 2 Information

ResultBuf contains a structure similar to the Level 1 structure, with the addition of the cbList field inserted before the name length field of the matched file object.

- 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

ResultBuf 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, fpGEAList contains the address of a GEA list, which defines the attribute names whose values are to be returned. fpGEAList is ignored. In case of error, oError contains the offset of the offending GEA entry. 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.

Following the EAOP structure is a structure similar to the Level 1 structure, with the addition of an FEAList structure inserted before the name length field for the matched file object.

- n output, this structure contains a packed set of records representing the directory entry and associated EAs for the matched file object. 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) : ASCIIZ name of EA.
- aValue (PSZ) : 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.

- ResultBufLen (USHORT) - input: Length of ResultBuf. SearchCount (PUSHORT) - input/output

:Address of the number of matching entries requested in ResultBuf. On return, this field contains the number of entries placed into ResultBuf.

- FileInfoLevel (USHORT): The level of file information required. A value of 1, 2, or 3 can be specified. The structures described in ResultBuf indicate the information returned for each of these levels.

Regardless of the level specified, a DosFindFirst2 request (and an associated DosFindNext request) always includes the information returned by level 1 as part of the information that is returned. However, when level 1 information is specifically requested, an inclusive search is made. That is, all normal file entries plus all entries matching any specified attributes are returned.

- Reserved (ULONG) - input : Reserved, must be set to zero.

Return Code

rc (USHORT) - return:Return code descriptions are:

- 0 NO_ERROR
- 2 ERROR_FILE_NOT_FOUND
- 3 ERROR_PATH_NOT_FOUND
- 6 ERROR_INVALID_HANDLE
- 18 ERROR_NO_MORE_FILES
- 26 ERROR_NOT_DOS_DISK
- 87 ERROR_INVALID_PARAMETER
- 108 ERROR_DRIVE_LOCKED
- 111 ERROR_BUFFER_OVERFLOW
- 113 ERROR_NO_MORE_SEARCH_HANDLES
- 206 ERROR_FILENAME_EXCED_RANGE
- 208 ERROR_META_EXPANSION_TOO_LONG
- 254 ERROR_INVALID_EA_NAME
- 255 ERROR_EA_LIST_ INCONSISTENT
- 275 ERROR_EAS_DIDNT_FIT

Remarks

DosFindFirst2 returns directory entries (up to the number requested in SearchCount) and extended attribute information for as many files or subdirectories whose names, attributes, and extended attributes match the specification, and whose information fits in ResultBuf. On output, SearchCount contains the actual number of directory entries returned.

DosFindNext uses the directory handle associated with DosFindFirst2 to continue the search started by the DosFindFirst2 request.

Any non-zero return code except ERROR_EAS_DIDNT_FIT indicates no handle has been allocated. This

includes such non-error indicators as ERROR_NO_MORE_FILES.

For programs running without the NEWFILES bit set, only 8.3 filename format names are returned. These names are changed to uppercase.

In the case of ERROR_EAS_DIDNT_FIT, a search handle is returned, and a subsequent call to DosFindNext will get the next matching entry in the directory. You may use DosQPathInfo to retrieve the EAs for the matching entry by using the EA arguments that were used for the DosFindFirst2 call and the name that was returned by DosFindFirst2.

In the case of ERROR_EAS_DIDNT_FIT, only information for the first matching entry is returned. This entry is the one whose EAs did not fit in the buffer. The information returned is in the format of that returned for InfoLevel 2. No further entries are returned in the buffer even if they could fit in the remaining space.

Family API Considerations

Some options operate differently in the DOS mode than in OS/2 mode. Therefore, the following restrictions apply to DosFindFirst when coding for the DOS mode:

DirHandle must always equal hex 0001H or FFFFH on the initial call to DosFindFirst. Subsequent calls to DosFindFirst must have a DirHandle of hex 0001H unless a DosFindClose had been issued. In this case, 0001H or FFFFH is allowed.

Bindings

C

```
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 _FILEFINDBUF {   /* findbuf */
  FDATE  fdateCreation;         /* file date of creation */
  FTIME  ftimeCreation;         /* file time of creation */
  FDATE  fdateLastAccess;       /* file date of last access */
  FTIME  ftimeLastAccess;       /* file time of last access */
  FDATE  fdateLastWrite;        /* file date of last write */
  FTIME  ftimeLastWrite;        /* file time of last write */
  ULONG  cbFile;                /* file end of data */
```

```
  ULONG  cbFileAlloc;           /* file allocation */
  USHORT attrFile;              /* file attribute */
  UCHAR  cchName;               /* length of ASCIIZ name string */
  CHAR   achName[CCHMAXPATHCOMP]; /* ASCIIZ name string */
} FILEFINDBUF;

typedef struct _FILEFINDBUF2 {   /* findbuf */
  FDATE  fdateCreation;         /* file date of creation */
  FTIME  ftimeCreation;         /* file time of creation */
  FDATE  fdateLastAccess;       /* file date of last access */
  FTIME  ftimeLastAccess;       /* file time of last access */
  FDATE  fdateLastWrite;        /* file date of last write */
  FTIME  ftimeLastWrite;        /* file time of last write */
  ULONG  cbFile;                /* file end of data */
  ULONG  cbFileAlloc;           /* file allocation */
  USHORT attrFile;              /* file attribute */
  ULONG  cbList;                /* level 2 only field (calculate size of
buffer) */
  UCHAR  cchName;               /* length of ASCIIZ name string */
  CHAR   achName[CCHMAXPATHCOMP]; /* ASCIIZ name string */
} FILEFINDBUF2;

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 = DosFindFirst2(FileName, DirHandle, Attribute, ResultBuf,
                           ResultBufLen, SearchCount, Reserved);

PSZ             FileName;      /* File path name */
PHDIR           DirHandle;     /* Directory search handle */
USHORT          Attribute;     /* Search attribute */
PVOID           ResultBuf;     /* Result buffer */
USHORT          ResultBufLen;  /* Result buffer length *
PUSHORT         SearchCount;   /* Number of entries to find */
USHORT          FileInfoLevel; /* File data required */
ULONG           0;             /* Reserved (must be zero) */

USHORT          rc;            /* return code */
```

MASM

```
FDATE   struc
  fdate_fs  dw  ?
FDATE   ends

FTIME   struc
  ftime_fs  dw  ?
FTIME   ends

FILEFINDBUF struc
  findbuf_fdateCreation   dw (size FDATE)/2 dup (?) ;file date of creation
  findbuf_ftimeCreation   dw (size FTIME)/2 dup (?) ;file time of creation
  findbuf_fdateLastAccess dw (size FDATE)/2 dup (?) ;file date of last
access
  findbuf_ftimeLastAccess dw (size FTIME)/2 dup (?) ;file time of last
access
  findbuf_fdateLastWrite  dw (size FDATE)/2 dup (?) ;file date of last write
  findbuf_ftimeLastWrite  dw (size FTIME)/2 dup (?) ;file time of last write
  findbuf_cbFile          dd  ? ;file end of data
  findbuf_cbFileAlloc     dd  ? ;file allocation
  findbuf_attrFile        dw  ? ;file attribute
  findbuf_cchName         db  ? ;length of ASCIIZ name string
  findbuf_achName         db  CCHMAXPATHCOMP dup (?) ;length of ASCIIZ name
string
FILEFINDBUF ends

FILEFINDBUF2 struc
  findbuf_fdateCreation   dw (size FDATE)/2 dup (?) ;file date of creation
  findbuf_ftimeCreation   dw (size FTIME)/2 dup (?) ;file time of creation
  findbuf_fdateLastAccess dw (size FDATE)/2 dup (?) ;file date of last
access
  findbuf_ftimeLastAccess dw (size FTIME)/2 dup (?) ;file time of last
access
  findbuf_fdateLastWrite  dw (size FDATE)/2 dup (?) ;file date of last write
  findbuf_ftimeLastWrite  dw (size FTIME)/2 dup (?) ;file time of last write
```

```
  findbuf_cbFile          dd  ? ;file end of data
  findbuf_cbFileAlloc     dd  ? ;file allocation
  findbuf_attrFile        dw  ? ;file attribute
  findbuf2_cbList         dd  ? ;level 2 only field (calculate size of
buffer)
  findbuf_achName         db  CCHMAXPATHCOMP dup (?) ;length of ASCIIZ name
string
  findbuf_achName         db  13 dup (?) ;ASCIIZ name string
FILEFINDBUF2 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  DosFindFirst2:FAR
INCL_DOSFILEMGR     EQU 1

PUSH@  ASCIIZ  FileName      ;File path name string
PUSH@  WORD    DirHandle     ;Directory search handle (returned)
PUSH   WORD    Attribute     ;Search attribute
PUSH@  OTHER   ResultBuf     ;Result buffer
PUSH   WORD    ResultBufLen  ;Result buffer length
PUSH@  WORD    SearchCount   ;Number of entries to find
PUSH   WORD    FileInfoLevel ;File data required
PUSH   DWORD   0             ;Reserved (must be zero)
CALL   DosFindFirst2
```

```
Returns WORD
```

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

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

Permanent link:
http://www.osfree.org/doku/doku.php?id=en:docs:fapi:dosfindfirst2

Last update: **2021/09/18 08:55**