 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

DosFindNext

This call locates the next set of directory entries that match the name specified in the previous DosFindFirst, DosFindFirst2, or DosFindNext call.

Syntax

```
DosFindNext (DirHandle, ResultBuf, ResultBufLen, SearchCount)
```

Parameters

- DirHandle (HDIR) - input : Handle associated with a previous DosFindFirst or DosFindNext function call.
- ResultBuf (PFILEFINDBUF) - output : Address of the directory search information structure. The information reflects the last DosClose or DosBufReset call.

It is possible, if the EA information for a file is 64K, that the system can never be able to return the full EA information for a file.

For the continuation of an FileInfoLevel 3 search, this buffer should contain input in the same format as a DosFindFirst2 FileInfoLevel 3 search.

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

- 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.

Return Code

rc (USHORT) - return

Return code descriptions are:

- 0 NO_ERROR
- 6 ERROR_INVALID_HANDLE
- 18 ERROR_NO_MORE_FILES
- 26 ERROR_NOT_DOS_DISK
- 87 ERROR_INVALID_PARAMETER
- 111 ERROR_BUFFER_OVERFLOW
- 275 ERROR_EAS_DIDNT_FIT

Remarks

The file name in FileName can contain global file name characters. If no more matching files are found, an error code is returned.

If an ERROR_BUFFER_OVERFLOW error is returned, further calls to DosFindNext will start the search from the same entry.

If an ERROR_EAS_DIDNT_FIT error is returned, then the buffer was too small to hold the EAs for the first matching entry being returned. A subsequent call to DosFindNext will get the next matching entry. This enables the search to continue if the EAs being returned are too big to fit in the buffer. 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 restriction applies to DosFindNext when coding for the DOS mode: DirHandle must always equal 1.

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 _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[13];           /* ASCIIZ name string */

} FILEFINDBUF;

#define INCL_DOSFILEMGR

USHORT  rc = DosFindNext(DirHandle, ResultBuf, ResultBufLen, SearchCount);

HDIR            DirHandle;     /* Directory handle */
PFILEFINDBUF    ResultBuf;     /* Result buffer */
USHORT          ResultBufLen;  /* Result buffer length */
```

```
PUSHORT           SearchCount;   /* Number of entries to find */

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

This example gets the 1st file in the current directory, and then gets the next file.

```
#define INCL_DOSFILEMGR

#define NORMAL_FILES 0
#define SEARCH_PATTERN "*.*"
#define FILE_ATTRIBUTE NORMAL_FILES
#define RESERVED 0L

HDIR        FindHandle;
FILEFINDBUF FindBuffer;
USHORT      FindCount;
USHORT      rc;

   FindHandle = 0x0001;
   FindCount = 1;

   if(!DosFindFirst(SEARCH_PATTERN,        /* File pattern */
                    &FindHandle,           /* Directory search handle */
                    FILE_ATTRIBUTE,        /* Search attribute */
                    &FindBuffer,           /* Result buffer */
                    sizeof(FindBuffer),    /* Result buffer length */
                    &FindCount,            /* # of entries to find */
                    RESERVED))             /* Reserved (must be zero) */
      rc = DosFindNext(FindHandle,         /* Directory handle */
                       &FindBuffer,        /* Result buffer */
                       sizeof(FindBuffer), /* Result buffer length */
                       &FindCount);        /* # of entries to find */
```

MASM Binding

```
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  13 dup (?) ;ASCIIZ name string

FILEFINDBUF ends

EXTRN  DosFindNext:FAR
INCL_DOSFILEMGR     EQU 1

PUSH   WORD    DirHandle     ;Directory search handle
PUSH@  OTHER   ResultBuf     ;Result buffer
PUSH   WORD    ResultBufLen  ;Result buffer length
PUSH@  WORD    SearchCount   ;Number of entries to find
CALL   DosFindNext

Returns WORD
```

Note

Text based on http://www.edm2.com/index.php/DosFindNext_(FAPI)

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
	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