

This call moves the read/write pointer in accordance with the type of move specified.

Syntax

DosChgFilePtr (FileHandle, Distance, MoveType, NewPointer)

Parameters

;FileHandle (HFILE) - input : Handle returned by a previous DosOpen call. ; Distance (LONG) - input : The offset to move, in bytes. ; MoveType (USHORT) - input : Method of moving. Specifies a location in the file from where Distance to move the read/write pointer starts. Values and their meanings are:
 'Value Definition' 0 Beginning of the file. 1 Current location of the read/write pointer. 2 End of the file. Use this method to determine a file's size. ; NewPointer (PULONG) - output : Address of the new pointer location.

Return Code

rc (USHORT) - return Return code descriptions are: * 0 NO_ERROR * 1 ERROR_INVALID_FUNCTION * 6 ERROR_INVALID_HANDLE

Remarks

The read/write pointer in a file is a signed 32-bit number. A negative value moves the pointer backward in the file. A positive value moves the pointer forward. DosChgFilePtr cannot be used to seek to a negative position in the file.

This call may not be used for a character device or pipe.

Example Code

C Binding

```
<PRE> #define INCL_DOSFILEMGR
```

```
USHORT rc = DosChgFilePtr(FileHandle, Distance, MoveType, NewPointer);
```

```
HFILE FileHandle; /* File handle */ LONG Distance; /* Distance to move in bytes */ USHORT MoveType;
/* Method of moving (0, 1, 2) */ PULONG NewPointer; /* New Pointer Location */
```

```
USHORT rc; /* return code */ </PRE> This example opens file test.dat, writes some data, and resets
the file pointer to the beginning of the file. <PRE> #define INCL_DOSFILEMGR
```

```
#define OPEN_FILE 0x01 #define CREATE_FILE 0x10 #define FILE_ARCHIVE 0x20 #define FILE_EXISTS
OPEN_FILE #define FILE_NOEXISTS CREATE_FILE #define DASD_FLAG 0 #define INHERIT 0x80
#define WRITE_THRU 0 #define FAIL_FLAG 0 #define SHARE_FLAG 0x10 #define ACCESS_FLAG 0x02
```

```
#define FILE_NAME "test.dat" #define FILE_SIZE 800L #define FILE_ATTRIBUTE FILE_ARCHIVE #define  
RESERVED 0L
```

```
HFILE FileHandle; USHORT Wrote; USHORT Action; PUSHORT Local PSZ FileData[100]; USHORT rc;
```

```
Action = 2;  
strcpy(FileData, "Data...");  
if(!DosOpen(FILE_NAME, /* File path name */  
            &FileHandle, /* File handle */  
            &Action, /* Action taken */  
            FILE_SIZE, /* File primary allocation */  
            FILE_ATTRIBUTE, /* File attribute */  
            FILE_EXISTS | FILE_NOEXISTS, /* Open function type */  
            DASD_FLAG | INHERIT | /* Open mode of the file */  
            WRITE_THRU | FAIL_FLAG |  
            SHARE_FLAG | ACCESS_FLAG,  
            RESERVED)) /* Reserved (must be zero) */  
    if(!DosWrite(FileHandle, /* File handle */  
                (PVOID) FileData, /* User buffer */  
                sizeof(FileData), /* Buffer length */  
                &Wrote)) /* Bytes written */  
        rc = DosChgFilePtr(FileHandle, /* File handle */  
                           MOVE_DIST, /* Distance to move in bytes */  
                           FILE_BEG, /* Method of moving */  
                           &Local); /* New pointer location */
```

</PRE>

MASM Binding

```
<PRE> EXTRN DosChgFilePtr:FAR INCL_DOSFILEMGR EQU 1
```

```
PUSH WORD FileHandle ;File handle PUSH DWORD Distance ;Distance to move in bytes PUSH WORD  
MoveType ;Method of moving (0, 1, 2) PUSH@ DWORD NewPointer ;New Pointer Location (returned)  
CALL DosChgFilePtr
```

Returns WORD </PRE>

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

Permanent link:
<https://www.osfree.org/doku/doku.php?id=en:docs:fapi:doschgfileptr&rev=1535290680>

Last update: **2018/08/26 13:38**

