



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](#)

## DosChgFilePtr

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

```
#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 */
```

This example opens file test.dat, writes some data, and resets the file pointer to the beginning of the file.

```
#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 */

```

## MASM Binding

```

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

## Note

Text based on <http://www.edm2.com/index.php/DosChgFilePtr>

Family API		
DOS	Process Manager	<a href="#">DosBeep</a> <a href="#">DosExit</a> <a href="#">DosSleep</a> <a href="#">DosExecPgm</a>
	File Manager	<a href="#">DosChDir</a> <a href="#">DosChgFilePtr</a> <a href="#">DosClose</a> <a href="#">DosDelete</a> <a href="#">DosDupHandle</a> <a href="#">DosMkDir</a> <a href="#">DosMove</a> <a href="#">DosQCurDir</a> <a href="#">DosQCurDisk</a> <a href="#">DosSetFileMode</a> <a href="#">DosOpen</a> <a href="#">DosQFileInfo</a> <a href="#">DosRead</a> <a href="#">DosQFileMode</a> <a href="#">DosQFSInfo</a> <a href="#">DosQVerify</a> <a href="#">DosRmDir</a> <a href="#">DosSelectDisk</a> <a href="#">DosFindClose</a> <a href="#">DosFindFirst</a> <a href="#">DosFindNext</a> <a href="#">DosSetFileInfo</a> <a href="#">DosSetVerify</a> <a href="#">DosWrite</a> <a href="#">DosFileLocks</a> <a href="#">DosSetFHandState</a> <a href="#">DosNewSize</a> <a href="#">DosBufReset</a> <a href="#">DosQFHandState</a> <a href="#">DosSetFSInfo</a>
	Memory Manager	<a href="#">DosFreeSeg</a> <a href="#">DosSubAlloc</a> <a href="#">DosSubFree</a> <a href="#">DosSubSet</a> <a href="#">DosAllocHuge</a> <a href="#">DosAllocSeg</a> <a href="#">DosReallocHuge</a> <a href="#">DosReallocSeg</a> <a href="#">DosGetHugeShift</a> <a href="#">DosCreateCSAlias</a>
	NLS	<a href="#">DosCaseMap</a> <a href="#">DosGetCtryInfo</a> <a href="#">DosGetDBCSEv</a> <a href="#">DosSetCtryCode</a> <a href="#">DosGetCollate</a> <a href="#">DosGetMessage</a> <a href="#">DosInsMessage</a> <a href="#">DosPutMessage</a>
	Date and Time	<a href="#">DosSetDateTime</a> <a href="#">DosGetDateTime</a>
	Devices	<a href="#">DosDevConfig</a> <a href="#">DosDevIOCtl</a> <a href="#">DosDevIOCtl2</a>
	Signals	<a href="#">DosHoldSignal</a> <a href="#">DosSetSigHandler</a>
	Misc	<a href="#">BadDynLink</a> <a href="#">DosGetEnv</a> <a href="#">DosGetMachineMode</a> <a href="#">DosGetVersion</a> <a href="#">DosError</a> <a href="#">DosErrClass</a> <a href="#">DosSetVec</a>
KBD		<a href="#">KbdCharIn</a> <a href="#">KbdFlushBuffer</a> <a href="#">KbdGetStatus</a> <a href="#">KbdSetStatus</a> <a href="#">KbdStringIn</a> <a href="#">KbdPeek</a>
VIO		<a href="#">VioGetBuf</a> <a href="#">VioGetConfig</a> <a href="#">VioGetCurPos</a> <a href="#">VioGetCurType</a> <a href="#">VioGetPhysBuf</a> <a href="#">VioReadCellStr</a> <a href="#">VioReadCharStr</a> <a href="#">VioScrollUp</a> <a href="#">VioScrollDn</a> <a href="#">VioScrollLf</a> <a href="#">VioScrollRt</a> <a href="#">VioScrUnLock</a> <a href="#">VioSetCurPos</a> <a href="#">VioSetCurType</a> <a href="#">VioSetMode</a> <a href="#">VioGetMode</a> <a href="#">VioShowBuf</a> <a href="#">VioWrtCellStr</a> <a href="#">VioWrtCharStr</a> <a href="#">VioWrtCharStrAtt</a> <a href="#">VioWrtNAttr</a> <a href="#">VioWrtNCell</a> <a href="#">VioWrtNChar</a> <a href="#">VioWrtTTY</a> <a href="#">VioScrLock</a> <a href="#">VioPopUp</a>
Tools		<a href="#">BIND</a>
Modules		<a href="#">DOSCALLS.DLL</a> <a href="#">VIOCALLS.DLL</a> <a href="#">KBDCALLS.DLL</a> <a href="#">MSG.DLL</a>
Libraries		<a href="#">API.LIB</a> <a href="#">OS2386.LIB</a> <a href="#">FAPI.LIB</a> <a href="#">DOSCALLS.LIB</a> <a href="#">SUBCALLS.LIB</a>

2018/08/25 15:05 · prokushev · [0 Comments](#)

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

Permanent link:  
<http://osfree.org./doku/doku.php?id=en:docs:fapi:doschgfileptr>

Last update: **2021/09/17 04:31**

