 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	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