 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

DosDupHandle

This call returns a new file handle for an open file, which refers to the same position in the file as the old file handle.

Syntax

```
DosDupHandle (OldFileHandle, NewFileHandle)
```

Parameters

- OldFileHandle (HFILE) - input : Current file handle.
- NewFileHandle (PHFILE) - input/output : Address of a Word. On input, values and their meanings are:
  - FFFFH - Allocate a new file handle and return it here.
  - <>FFFFH - Assign this value as the new file handle. A valid value is any of the handles assigned to standard I/O, or the handle of a file currently opened by the process.
- On output, a value of FFFFH returns a value for NewFileHandle, allocated by OS/2.

Return Code

rc (USHORT) - return

Return code descriptions are:

- 0 NO_ERROR
- 4 ERROR_TOO_MANY_OPEN_FILES
- 6 ERROR_INVALID_HANDLE
- 114 ERROR_INVALID_TARGET_HANDLE

Remarks

Duplicating the handle duplicates and ties all handle-specific information between OldFileHandle and

NewFileHandle. For example, if you move the read/write pointer of either handle by a DosRead, DosWrite, or DosChgFilePtr function call, the pointer for the other handle is also changed.

The valid values for NewFileHandle include the following handles for standard I/O, which are always available to the process:

- 0000H Standard input
- 0001H Standard output
- 0002H Standard error.

If a file handle value of a currently open file is specified in NewFileHandle, the file handle is closed before it is redefined as the duplicate of OldFileHandle. Avoid using arbitrary values for NewFileHandle.

Issuing a DosClose against a file handle does not affect the duplicate handle.

Example Code

C Binding

```
#define INCL_DOSFILEMGR

USHORT  rc = DosDupHandle(OldFileHandle, NewFileHandle);

HFILE           OldFileHandle; /* Existing file handle */
PHFILE          NewFileHandle; /* New file handle (returned) */

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

This example opens a file, creates a second file handle, then closes the file with the second handle.

```
#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;
  HFILE   NewHandle
  USHORT  Wrote;
  USHORT  Action;
  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) */
      rc = DosDupHandle(FileHandle,      /* Existing file handle */
                        &NewHandle);     /* New file handle */
```

MASM Binding

```
EXTRN  DosDupHandle:FAR
INCL_DOSFILEMGR     EQU 1

PUSH   WORD    OldFileHandle ;Existing file handle
PUSH@  WORD    NewFileHandle ;New file handle (returned)
CALL   DosDupHandle
```

Returns WORD

Note

Text based on http://www.edm2.com/index.php/DosDupHandle_(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 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