



# 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

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

```
USHORT rc = DosDupHandle(OldFileHandle, NewFileHandle);
```

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

```
USHORT rc; /* return code */ </PRE> This example opens a file, creates a second file handle, then closes the file with the second handle.
```

```
<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; 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 */
```

```
</PRE>
```

## MASM Binding

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

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

```
Returns WORD </PRE>
```

# Note

Text based on [http://www.edm2.com/index.php/DosDupHandle\\_\(FAPi\)](http://www.edm2.com/index.php/DosDupHandle_(FAPi))

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> <a href="#">DosShutdown</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="#">FAP.LIB</a> <a href="#">DOSCALLS.LIB</a> <a href="#">SUBCALLS.LIB</a>	

2018/08/25 15:05 · prokushev · 0 Comments

From:

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

Permanent link:

<https://cocorico.osfree.org/doku/doku.php?id=en:docs:fapi:dosduphandle&rev=1535294379>

Last update: **2018/08/26 14:39**

