



# DosBufReset

This call flushes a requesting process's cache buffers for a specific file handle or for all file handles attached to that process. This call is also used with the handle of a named pipe to synchronize a dialog between communicating processes.

## Syntax

```
DosBufReset (FileHandle)
```

## Parameters

- FileHandle (HFILE) - input : File handle whose buffers are to be flushed. If FileHandle = 0xFFFFH, all of the process's file handles are flushed.

## Return Code

```
rc (USHORT) - return
```

Return code descriptions are:

- 0 NO\_ERROR
- 2 ERROR\_FILE\_NOT\_FOUND
- 5 ERROR\_ACCESS\_DENIED
- 6 ERROR\_INVALID\_HANDLE

## Remarks

Upon issuing DosBufReset for a file handle, the file's buffers are flushed to disk and its directory entry updated as if the file had been closed; however the file remains in an open state.

Usage of this call to write out all files belonging to the requesting process should be administered with caution. When the files reside on removable media (diskettes), a call to DosBufReset could have the undesirable effect of requiring the user to insert and remove a large number of diskettes.

## Named Pipe Considerations

Issuing DosBufReset for a named pipe performs an operation that is analogous to forcing the buffer cache to disk. The request blocks the calling process at one end of the pipe until all data it has written has been read at the other end of the pipe.

## Example Code

## C Binding

```
#define INCL_DOSFILEMGR

USHORT rc = DosBufReset(FileHandle);

HFILE      FileHandle;    /* File handle */

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

This example opens a file, writes some data to the file's buffer, then flushes the file's system buffer.

```
#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;
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 = DosBufReset(FileHandle);    /* File handle */
    
```

### MASM Binding

```

EXTRN  DosBufReset:FAR
INCL_DOSFILEMGR    EQU 1

PUSH   WORD        FileHandle    ;File handle
CALL   DosBufReset
    
```

Returns WORD

## Note

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

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 DosDevIOct1 DosDevIOct2
	Signals	DosHoldSignal DosSetSigHandler
	Misc	BadDynLink DosGetEnv DosGetMachineMode DosGetVersion DosError DosErrClass DosSetVec
KBD		KbdCharIn KbdFlushBuffer KbdGetStatus KbdSetStatus KbdStringIn KbdPeek

<b>Family API</b>	
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.ru/doku/> - **osFree wiki**

Permanent link:  
<http://osfree.ru/doku/doku.php?id=en:docs:fapi:dosbufreset&rev=1607251171>

Last update: **2020/12/06 10:39**

