



# DosFreeSeg

This call deallocates a memory segment.

## Syntax

```
DosFreeSeg (Selector)
```

## Parameters

- Selector ([SEL](#)) - input : Selector of the segment to be freed.

## Return Code

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

Return code descriptions are:

- 0 NO\_ERROR
- 5 ERROR\_ACCESS\_DENIED
- 212 ERROR\_LOCKED

## Remarks

DosFreeSeg frees selectors to segments returned by allocation calls to [DosAllocSeg](#), [DosAllocShrSeg](#), and [DosAllocHuge](#). In addition, DosFreeSeg frees a selector returned by a call to [DosCreateCSAlias](#). If a CS alias selector has been created for a data segment by a call to [DosCreateCSAlias](#), the CS alias selector is still valid after the segment's data selector has been freed.

When allocated memory is shared, all selectors to the shared memory must be freed before the memory is deallocated. For example, if memory allocated by [DosAllocSeg](#) or [DosAllocHuge](#) has been given to another process with [DosGiveSeg](#), the giver usually frees its selector by a call to [DosFreeSeg](#). The recipient, in turn, frees the selector passed to it, after it has accessed the shared memory with [DosGetSeg](#).

DosFreeSeg decrements the reference count for named shared segments allocated by [DosAllocShrSeg](#). Access to the segment with [DosGetShrSeg](#) increments this count. When the count is 0, the memory is deallocated.

## Family API Considerations

Some options operate differently in the DOS mode than in OS/2 mode. Therefore, the following restriction applies to DosFreeSeg when coding for the DOS mode:

If DosFreeSeg is issued on a CSAliased segment it deallocates the associated memory. This is inconsistent with the OS/2 mode, because DosFreeSeg must be performed on both the original and CSAliased selectors.

### Example Code

## C Binding

```
#define INCL_DOSMEMMGR

USHORT rc = DosFreeSeg(Selector);

SEL      Selector;      /* Selector */

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

This example allocates a segment of 30,250 bytes and then discards the segment.

```
#define INCL_DOSMEMMGR

#define NUMBER_OF_BYTES 30250
#define ALLOC_FLAG SEG_GETTABLE

SEL      Selector;
USHORT   rc;

    rc = DosAllocSeg(NUMBER_OF_BYTES,      /* # of bytes requested */
                    &Selector,          /* Selector allocated */
                    ALLOC_FLAG);        /* Allocation flags */
rc = DosFreeSeg(Selector);              /* Segment selector */
```

## MASM Binding

```
EXTRN  DosFreeSeg:FAR
INCL_DOSMEMMGR      EQU 1

PUSH   WORD      Selector      ;Selector
CALL   DosFreeSeg
```

Returns WORD

# Note

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

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

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

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

Permanent link: <https://ftp.osfree.org/doku/doku.php?id=en:docs:fapi:dosfreeseq&rev=1535906135>

Last update: **2018/09/02 16:35**

