

This call changes the size of memory originally allocated by DosAllocHuge.

## Syntax

DosReallocHuge (NumSeg, Size, Selector)

## Parameters

;NumSeg (USHORT) - input : Number of 65536 byte segments requested. ;Size (USHORT) - input : Number of bytes requested in the last non-65536 byte segment. A value of 0 indicates none. ;Selector (SEL) - input : Selector returned on a previous DosAllocHuge.

## Return Code

rc (USHORT) - return Return code descriptions are:  
\* 0 NO\_ERROR  
\* 8 ERROR\_NOT\_ENOUGH\_MEMORY  
\* 87 ERROR\_INVALID\_PARAMETER

## Remarks

DosReallocHuge is called to change the size of unshared or shared huge memory allocated by DosAllocHuge. The selector used for this call must be the one returned by the DosAllocHuge request.

Normally, segments allocated as shared (AllocFlags bits 0 and 1 were set) cannot be decreased in size. However, if AllocFlags bit 3 was also set, the shared segment's size can be decreased.

DosReallocHuge is also called to reallocate a segment allocated as discardable (AllocFlags bit 2 set) after the segment is discarded by the system. The call to DosReallocHuge automatically locks the segment for access by the caller, the same as if a DosLockSeg had been issued.

'Note:' This request may be issued from privilege level 2 or 3. However, only a privilege level 3 huge segment is valid.

## Family API Considerations

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

The requested Size value is rounded up to the next paragraph (16-byte).

## Example Code

## C Binding

```
<PRE> #define INCL_DOSMEMMGR

USHORT rc = DosReallocHuge(NumSeg, Size, Selector);

USHORT NumSeg; /* Number of 65536-byte segments

                           requested */

USHORT Size; /* Number of bytes in last segment */ SEL Selector; /* Selector */

USHORT rc; /* return code */ </PRE>
```

## MASM Binding

```
<PRE> EXTRN DosReallocHuge:FAR INCL_DOSMEMMGR EQU 1
```

```
PUSH WORD NumSeg ;Number of 65536-byte segments requested PUSH WORD Size ;Number of bytes
in last segment PUSH WORD Selector ;Selector CALL DosReallocHuge
```

Returns WORD </PRE>

## Note

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

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="#">DosSet FileMode</a> <a href="#">DosOpen</a> <a href="#">DosQFileInfo</a> <a href="#">DosRead</a> <a href="#">DosQ FileMode</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="#">DosSet FileInfo</a> <a href="#">DosSet Verify</a> <a href="#">DosWrite</a> <a href="#">DosFileLocks</a> <a href="#">DosSet FHand State</a> <a href="#">DosNew Size</a> <a href="#">DosBufReset</a> <a href="#">DosQF Hand State</a> <a href="#">DosSet FSinfo</a>
	Memory Manager <a href="#">DosFreeSeg</a> <a href="#">DosSubAlloc</a> <a href="#">DosSubFree</a> <a href="#">DosSubSet</a> <a href="#">DosAlloc Huge</a> <a href="#">DosAlloc Seg</a> <a href="#">DosRealloc Huge</a> <a href="#">DosRealloc Seg</a> <a href="#">DosGet Huge Shift</a> <a href="#">DosCreate CS Alias</a>
	NLS <a href="#">DosCaseMap</a> <a href="#">DosGet Ctry Info</a> <a href="#">DosGet DBCSEv</a> <a href="#">DosSet Ctry Code</a> <a href="#">DosGet Collate</a> <a href="#">DosGet Message</a> <a href="#">DosIns Message</a> <a href="#">DosPut Message</a>
	Date and Time <a href="#">DosSet Date Time</a> <a href="#">DosGet Date Time</a>
	Devices <a href="#">DosDevConfig</a> <a href="#">DosDevIOCtl</a> <a href="#">DosDevIOCtl2</a>
	Signals <a href="#">DosHold Signal</a> <a href="#">DosSet Sig Handler</a>
	Misc <a href="#">BadDynLink</a> <a href="#">DosGet Env</a> <a href="#">DosGet Machine Mode</a> <a href="#">DosGet Version</a> <a href="#">DosError</a> <a href="#">DosErr Class</a> <a href="#">DosSet Vec</a>
KBD <a href="#">KbdCharIn</a> <a href="#">KbdFlushBuffer</a> <a href="#">KbdGet Status</a> <a href="#">KbdSet Status</a> <a href="#">KbdStringIn</a> <a href="#">KbdPeek</a>	

Family API	
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	BIND
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.su/doku/> - osFree wiki



Permanent link:

<http://osfree.su/doku/doku.php?id=en:docs:fapi:dosreallochuge&rev=1535649648>

Last update: **2018/08/30 17:20**