VirtualFreeEx function

Releases, decommits, or releases and decommits a region of memory within the virtual address space of a specified process.

Syntax

C++

```
BOOL WINAPI VirtualFreeEx(
  _In_ HANDLE hProcess,
  _In_ LPVOID lpAddress,
  _In_ SIZE_T dwSize,
  _In_ DWORD  dwFreeType
);
```

Parameters

hProcess [in]

A handle to a process. The function frees memory within the virtual address space of the process.

The handle must have the **PROCESS_VM_OPERATION** access right. For more information, see Process Security and Access Rights.

lpAddress [in]

A pointer to the starting address of the region of memory to be freed.

If the *dwFreeType* parameter is **MEM_RELEASE**, *lpAddress* must be the base address returned by the **VirtualAllocEx** function when the region is reserved.

dwSize [in]

The size of the region of memory to free, in bytes.

If the *dwFreeType* parameter is **MEM_RELEASE**, *dwSize* must be 0 (zero). The function frees the entire region that is reserved in the initial allocation call to **VirtualAllocEx**.

If *dwFreeType* is **MEM_DECOMMIT**, the function decommits all memory pages that contain one or more bytes in the range from the *lpAddress* parameter to `(lpAddress+dwSize)`. This means, for example, that a 2-byte region of memory that straddles a page boundary causes both pages to be decommitted. If *lpAddress* is the base address returned by **VirtualAllocEx** and *dwSize* is 0 (zero), the function decommits the entire region that is allocated by **VirtualAllocEx**. After that, the entire region is in the reserved state.

dwFreeType [in]

The type of free operation. This parameter can be one of the following values.

Value	Meaning
MEM_DECOMMIT 0x4000	Decommits the specified region of committed pages. After the operation, the pages are in the reserved state. The function does not fail if you attempt to decommit an uncommitted page. This means that you can decommit a range of pages without first determining their current commitment state. Do not use this value with **MEM_RELEASE**. The **MEM_DECOMMIT** value is not supported when the *lpAddress* parameter provides the base address for an enclave.
MEM_RELEASE 0x8000	Releases the specified region of pages. After the operation, the pages are in the free state. If you specify this value, *dwSize* must be 0 (zero), and *lpAddress* must point to the base address returned by the VirtualAllocEx function when the region is reserved. The function fails if either of these conditions is not met. If any pages in the region are committed currently, the function first decommits, and then releases them. The function does not fail if you attempt to release pages that are in different states, some reserved and some committed. This means that you can release a range of pages without first determining the current commitment state. Do not use this value with **MEM_DECOMMIT**.

Return value

If the function succeeds, the return value is a nonzero value.

If the function fails, the return value is 0 (zero). To get extended error information, call GetLastError.

Remarks

Each page of memory in a process virtual address space has a Page State. The **VirtualFreeEx** function can decommit a range of pages that are in different states, some committed and some uncommitted. This means

that you can decommit a range of pages without first determining the current commitment state of each page. Decommitting a page releases its physical storage, either in memory or in the paging file on disk.

If a page is decommitted but not released, its state changes to reserved. Subsequently, you can call VirtualAllocEx to commit it, or **VirtualFreeEx** to release it. Attempting to read from or write to a reserved page results in an access violation exception.

The **VirtualFreeEx** function can release a range of pages that are in different states, some reserved and some committed. This means that you can release a range of pages without first determining the current commitment state of each page. The entire range of pages originally reserved by VirtualAllocEx must be released at the same time.

If a page is released, its state changes to free, and it is available for subsequent allocation operations. After memory is released or decommitted, you can never refer to the memory again. Any information that may have been in that memory is gone forever. Attempts to read from or write to a free page results in an access violation exception. If you need to keep information, do not decommit or free memory that contains the information.

The **VirtualFreeEx** function can be used on an AWE region of memory and it invalidates any physical page mappings in the region when freeing the address space. However, the physical pages are not deleted, and the application can use them. The application must explicitly call FreeUserPhysicalPages to free the physical pages. When the process is terminated, all resources are automatically cleaned up.

To delete an enclave when you finish using it, specify the following values:

- The base address of the enclave for the *lpAddress* parameter.
- 0 for the *dwSize* parameter.
- **MEM_RELEASE** for the *dwFreeType* parameter. The **MEM_DECOMMIT** value is not supported for enclaves.

Requirements

Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Header	WinBase.h (include Windows.h)
Library	Kernel32.lib
DLL	Kernel32.dll

See also

Memory Management Functions
Virtual Memory Functions
VirtualAllocEx