

InitializeCriticalSection function

Initializes a critical section object.

Syntax

C++

```
void WINAPI InitializeCriticalSection(  
    _Out_ LPCRITICAL_SECTION lpCriticalSection  
);
```

Parameters

lpCriticalSection [out]
A pointer to the critical section object.

Return value

This function does not return a value.

Windows Server 2003 and Windows XP: In low memory situations, **InitializeCriticalSection** can raise a **STATUS_NO_MEMORY** exception. This exception was eliminated starting with Windows Vista.

Remarks

The threads of a single process can use a critical section object for mutual-exclusion synchronization. There is no guarantee about the order in which threads will obtain ownership of the critical section, however, the system will be fair to all threads.

The process is responsible for allocating the memory used by a critical section object, which it can do by declaring a variable of type **CRITICAL_SECTION**. Before using a critical section, some thread of the process must initialize the object.

After a critical section object has been initialized, the threads of the process can specify the object in the [EnterCriticalSection](#), [TryEnterCriticalSection](#), or [LeaveCriticalSection](#) function to provide mutually exclusive access to a shared resource. For similar synchronization between the threads of different processes, use a mutex object.

A critical section object cannot be moved or copied. The process must also not modify the object, but must treat it as logically opaque. Use only the critical section functions to manage critical section objects. When you have finished using the critical section, call the [DeleteCriticalSection](#) function.

A critical section object must be deleted before it can be reinitialized. Initializing a critical section that has already been initialized results in undefined behavior.

Requirements

Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Header	WinBase.h on Windows XP, Windows Server 2003, Windows Vista, Windows 7, Windows Server 2008, and Windows Server 2008 R2 (include Windows.h); Synchapi.h on Windows 8 and Windows Server 2012

Library	Kernel32.lib
<hr/>	
DLL	Kernel32.dll
<hr/>	

See also

- [CreateMutex](#)
- [Critical Section Objects](#)
- [DeleteCriticalSection](#)
- [EnterCriticalSection](#)
- [InitializeCriticalSectionAndSpinCount](#)
- [LeaveCriticalSection](#)
- [Synchronization Functions](#)
- [TryEnterCriticalSection](#)

Community Additions