ReleaseSemaphore function

Increases the count of the specified semaphore object by a specified amount.

Syntax

C++

```
BOOL WINAPI ReleaseSemaphore(
  _In_      HANDLE hSemaphore,
  _In_      LONG   lReleaseCount,
  _Out_opt_ LPLONG lpPreviousCount
);
```

Parameters

hSemaphore [in]

A handle to the semaphore object. The CreateSemaphore or OpenSemaphore function returns this handle.

This handle must have the **SEMAPHORE_MODIFY_STATE** access right. For more information, see Synchronization Object Security and Access Rights.

lReleaseCount [in]

The amount by which the semaphore object's current count is to be increased. The value must be greater than zero. If the specified amount would cause the semaphore's count to exceed the maximum count that was specified when the semaphore was created, the count is not changed and the function returns **FALSE**.

lpPreviousCount [out, optional]

A pointer to a variable to receive the previous count for the semaphore. This parameter can be **NULL** if the previous count is not required.

Return value

If the function succeeds, the return value is nonzero.

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

Remarks

The state of a semaphore object is signaled when its count is greater than zero and nonsignaled when its count is equal to zero. The process that calls the CreateSemaphore function specifies the semaphore's initial count. Each time a waiting thread is released because of the semaphore's signaled state, the count of the semaphore is decreased by one.

Typically, an application uses a semaphore to limit the number of threads using a resource. Before a thread uses the resource, it specifies the semaphore handle in a call to one of the wait functions. When the wait function returns, it decreases the semaphore's count by one. When the thread has finished using the resource, it calls **ReleaseSemaphore** to increase the semaphore's count by one.

Another use of **ReleaseSemaphore** is during an application's initialization. The application can create a semaphore with an initial count of zero. This sets the semaphore's state to nonsignaled and blocks all threads from accessing the protected resource. When the application finishes its initialization, it uses **ReleaseSemaphore** to increase the count to its maximum value, to permit normal access to the protected resource.

It is not possible to reduce the semaphore object count using **ReleaseSemaphore**, because *lReleaseCount* cannot be a negative number. To temporarily restrict or reduce access, create a loop in which you call the WaitForSingleObject function with a time-out interval of zero until the semaphore count has been reduced sufficiently. (Note that other threads can reduce the count while this loop is being executed.) To restore access, call **ReleaseSemaphore** with the release count equal to the number of times **WaitForSingleObject** was called in the loop.

Windows Phone 8: This API is supported.

Windows Phone 8.1: This API is supported.

Examples

For an example that uses **ReleaseSemaphore**, see Using Semaphore Objects.

Requirements

Minimum supported client	Windows XP [desktop apps | Windows Store apps]
Minimum supported server	Windows Server 2003 [desktop apps | Windows Store apps]
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
DLL	Kernel32.dll

See also

CreateSemaphore
OpenSemaphore
Semaphore Objects
Synchronization Functions

Community Additions

VB9 vs VB.NET

These declarations seem to differ (one has a paramater as int, the other as intptr)- can the author create a single vb.net declaration?

Thomas Lee
5/18/2009

C# syntax

```
[DllImport("kernel32.dll", SetLastError=true)]
internal static extern bool ReleaseSemaphore(IntPtr handle, int releaseCount, out int previousCount);
```

dmex
5/7/2009

vb.net syntax

```
<DllImport("kernel32.dll", SetLastError:=True)> _
Public Shared Function ReleaseSemaphore(ByVal handle As IntPtr, ByVal releaseCount As Integer, <Out> ByRef previousCount As Integer) As Boolean
End Function
```

dmex
5/7/2009

VB9 syntax

Public Declare Function ReleaseSemaphore Lib "kernel32" Alias "ReleaseSemaphore" (ByVal hSemaphore As Integer, ByVal lReleaseCount As Integer, ByVal lpPreviousCount As Integer) As Integer

dmex

5/7/2009