

# Función FreeLibrary (libloaderapi.h)

Artículo • 13/10/20212 minutos para leer

Libera el módulo de biblioteca de vínculos dinámicos (DLL) cargado y, si es necesario, reduce su recuento de referencias. Cuando el recuento de referencias llega a cero, el módulo se descarga del espacio de direcciones del proceso de llamada y el identificador ya no es válido.

## Sintaxis

C++

```
BOOL FreeLibrary(  
    [in] HMODULE hLibModule  
);
```

## Parámetros

[in] hLibModule

Un identificador para el módulo de biblioteca cargado. La función [LoadLibrary](#) , [LoadLibraryEx](#) , [GetModuleHandle](#) o [GetModuleHandleEx](#) devuelve este identificador.

## Valor devuelto

Si la función tiene éxito, el valor devuelto es distinto de cero.

Si la función falla, el valor de retorno es cero. Para obtener información de error extendida, llame a la función [GetLastError](#) .

## Observaciones

El sistema mantiene un conteo de referencia por proceso para cada módulo cargado. Un módulo que se cargó en la inicialización del proceso debido a la vinculación dinámica en el tiempo de carga tiene un recuento de referencia de uno. El recuento de referencias de un módulo se incrementa cada vez que se carga el módulo mediante una llamada a [LoadLibrary](#) . El recuento de referencias también se incrementa mediante una

llamada a [LoadLibraryEx](#) , a menos que el módulo se cargue por primera vez y se cargue como un archivo de datos o imagen.

El recuento de referencias se reduce cada vez que se llama a la función **FreeLibrary** o [FreeLibraryAndExitThread](#) para el módulo. Cuando el recuento de referencias de un módulo llega a cero o el proceso finaliza, el sistema descarga el módulo del espacio de direcciones del proceso. Antes de descargar un módulo de biblioteca, el sistema permite que el módulo se separe del proceso llamando a la función [DllMain](#) del módulo , si tiene una, con el valor `DLL_PROCESS_DETACH`. Hacerlo le da al módulo de la biblioteca la oportunidad de limpiar los recursos asignados en nombre del proceso actual. Después de que regresa la función de punto de entrada, el módulo de biblioteca se elimina del espacio de direcciones del proceso actual.

It is not safe to call **FreeLibrary** from [DllMain](#). For more information, see the Remarks section in [DllMain](#).

Calling **FreeLibrary** does not affect other processes that are using the same module.

Use caution when calling **FreeLibrary** with a handle returned by [GetModuleHandle](#). The **GetModuleHandle** function does not increment a module's reference count, so passing this handle to **FreeLibrary** can cause a module to be unloaded prematurely.

A thread that must unload the DLL in which it is executing and then terminate itself should call [FreeLibraryAndExitThread](#) instead of calling **FreeLibrary** and **ExitThread** separately. Otherwise, a race condition can occur. For details, see the Remarks section of [FreeLibraryAndExitThread](#).

## Examples

For an example, see [Using Run-Time Dynamic Linking](#).

## Requirements

Minimum supported client	Windows XP [desktop apps   UWP apps]
Minimum supported server	Windows Server 2003 [desktop apps   UWP apps]
Target Platform	Windows
Header	libloaderapi.h (include Windows.h)
Library	Kernel32.lib

DLL	Kernel32.dll

# See also

- [DllMain](#)
- [Dynamic-Link Library Functions](#)
- [FreeLibraryAndExitThread](#)
- [GetModuleHandle](#)
- [GetModuleHandleEx](#)
- [LoadLibrary](#)
- [Run-Time Dynamic Linking](#)