

How am I supposed to free the memory the system allocates in the SetPrivateObjectSecurity function?

devblogs.microsoft.com/oldnewthing/20170727-00

July 27, 2017



Raymond Chen

A customer noted that [the SetPrivateObjectSecurity function](#) updates a pointer provided by the `ObjectsSecurityDescriptor` parameter. Since it may allocate a new security descriptor, that means that it needs to deallocate the old one. But what function does it use to free the old one? After all, the allocation function must match the deallocation function. Similarly, how should the new security descriptor be freed? (I say “similarly” because the two answers had better be the same!)

The system allocates and frees the security descriptor from the process heap, as reported by the `GetProcessHeap` function. The allocation function is `HeapAlloc` and the deallocation function is `HeapFree`. That means that the security descriptor you pass in must have been allocated with

```
SecurityDescriptor = HeapAlloc(GetProcessHeap(), flags, size);
```

and then you pass the pointer like this:

```
SetPrivateObjectSecurity(..., &SecurityDescriptor, ...);  
// or  
SetPrivateObjectSecurityEx(..., &SecurityDescriptor, ...);
```

and after the `SetPrivateObjectSecurity` function is done, you must free the memory with

```
HeapFree(GetProcessHeap(), SecurityDescriptor);
```

I wrote this post the same day that I submitted the change request to add this essential information to the documentation. We’ll see who wins.

Raymond Chen

Follow

