Under what conditions can I modify the memory that I received in the form a STGMEDIUM?

devblogs.microsoft.com/oldnewthing/20220701-00

July 1, 2022

Raymond Chen

A customer was looking to optimize their use of data that they received from a data object in the form of a `STGMEDIUM`. Right now, they are making a copy of the `hGlobal` in the `STGMEDIUM` and modifying the copy. But that memory block could be quite large. Is it possible for them to just modify the original `hGlobal`? What are the ownership rules for the contents of a `STGMEDIUM`?

The rule is that you call `ReleaseStgMedium` when you are finished with a `STGMEDIUM`. If you look at the details of the `ReleaseStgMedium` function, it behaves in one of two modes, depending on whether the `punkForRelease` member is null.

Medium	`punkForRelease`		
	`nullptr`	Not `nullptr` (perform both columns)	
`TYMED_HGLOBAL`	`GlobalFree`	Nothing	`punkForRelease->Release()`
`TYMED_GDI`	`DeleteObject`	Nothing	`punkForRelease->Release()`
`TYMED_ENHMF`	`DeleteEnhMetaFile`	Nothing	`punkForRelease->Release()`
`TYMED_MFPICT`	`DeleteMetaFile` + `GlobalFree`	Nothing	`punkForRelease->Release()`
`TYMED_FILE`	`DeleteFile` + `CoTaskMemFree`	`CoTaskMemFree`	`punkForRelease->Release()`
`TYMED_ISTREAM`	`IStream::Release`	`IStream::Release`	`punkForRelease->Release()`
`TYMED_ISTORAGE`	`IStorage::Release`	`IStorage::Release`	`punkForRelease->Release()`

You can see that the model is that a null `punkForRelease` means that the medium is owned by the code that possesses the `STGMEDIUM`, whereas a non-null `punkForRelease` means that the medium is controlled by the `punkForRelease`. (In the `TYMED_FILE` case, the logical medium is the file on disk; the file name is always freed. And the distinction is irrelevant for `IStream` and `IStorage` cases, since the interface pointer is being released either way.)

This means that if the `punkForRelease` is null, you can just treat the medium as if you owned it. In the null `punkForRelease` case, all `ReleaseStgMedium` is going to do is free the `hGlobal`. You can rescue that memory just before it reaches the incinerator and use it for whatever purpose you like. It was going to be destroyed anyway.

On the other hand, if the `punkForRelease` is non-null, then you need to copy the memory and modify your copy, because the `hGlobal` is owned by the `punkForRelease`.[1]

[1] The non-null `punkForRelease` case typically occurs when the data object that provided the `STGMEDIUM` wants to cache the data across multiple calls to `GetData`. It creates the data once and returns the handle to each caller, but setting the cache as the `punkForRelease`. (In most cases, the data object acts as its own cache, so it passes itself as the `punkForRelease`.)

Raymond Chen

Follow