What was so horrible about the FORMAT_MESSAGE_ALLOCATE_BUFFER flag that the Windows Store disallowed it for so long?

devblogs.microsoft.com/oldnewthing/20200806-00

August 6, 2020

Raymond Chen

Last time, we learned about the tumultuous history of the `FORMAT_MESSAGE_ALLOCATE_BUFFER` flag in Windows Store UWP apps.

But why was this flag disallowed for so long?

It's nothing particularly profound. Rather, it was just bad luck.

The buffer allocated by the `FORMAT_MESSAGE_ALLOCATE_BUFFER` flag needs to be freed by calling `LocalFree`, but `LocalFree` was not one of the functions that can be called from a Windows Store app.

Why not?

Because `LocalAlloc` and `LocalFree` are legacy functions that hang around for backward compatibility with 16-bit Windows. New programs shouldn't be using them. It's not like your new program needs to be backward compatible with 16-bit Windows 3.1.

But this left the `FORMAT_MESSAGE_ALLOCATE_BUFFER` flag in a bit of a pickle, because despite being something new for Win32, the flag continues to use that old and busted legacy function for memory allocation.

There was some discussion within the team about how to address the problem. One school of thought was to document enough of the internals of the `LocalFree` function so that you could call `HeapFree` to free it. You can see remnants of this approach in the comments of the `winbase.h` header file:

```
//
// FORMAT_MESSAGE_ALLOCATE_BUFFER requires use of HeapFree
//
```

For a while, there was also this paragraph of the `FORMAT_MESSAGE_ALLOCATE_BUFFER` documentation that says a bunch of stuff in a rather confusing way.

> **Windows 10**: **LocalAlloc()** has different options: **LMEM_FIXED**, and **LMEM_MOVABLE**. **FormatMessage()** uses **LMEM_FIXED**, so **HeapFree** can be used. If **LMEM_MOVABLE** is used, **HeapFree** cannot be used.

(Fortunately, the confusing paragraph isn't there any more.)

The decision that won the day was to accept that legacy code will never die. The team just held their nose and added `LocalAlloc` and `LocalFree` to the list of functions that are permitted to be called by Windows Store app.

But please, promise to use it only for situations that absolutely require it for compatibility purposes, like freeing the message string allocated by `FormatMessage`. Don't use it as your go-to memory allocation function.

Raymond Chen

Follow