Limitations on DLL resources in Windows 95

devblogs.microsoft.com/oldnewthing/20030807-00

August 7, 2003

Raymond Chen

Ancient history lesson.

When Win9x loads a 32-bit DLL, it creates a shadow 16-bit DLL so 16-bit code (like USER) can access resources in it.

The shadow DLL is effectively a resource-only 16-bit DLL, created by taking the 32-bit resources and converting them to 16-bit format. If the resources cannot be converted to 16-bit format, the DLL will not load.

The 16-bit resource file format specifies resource sizes by combining a DLL-wide shift value with a 16-bit per-resource scaled size. So, for example, if the shift value were 2, and the per-resource scaled size were 8, then the actual resource size would be 8 << 2 = 32.

Windows 95 has a bug in the way it calculates the scaled size.

If the Windows 95 kernel decided that it needed to use a nonzero shift value because the 32-bit DLL contains a resource larger than 64K, it scaled the 32-bit values down to 16-bit values and rounded *down* rather than up. So, for example, if a resources were 65537 bytes in size and the shift factor were 1, then the scaled-down value would be 65537 >> 1 = 32768. After scaling back up, the result would be 32768 >> 1 = 65536. Notice that the value is too small; the last byte of the resource has been truncated.

Consequently, if you have a 32-bit DLL with resources 64K or larger, you must pad those resources to prevent this truncation from happening. In the above example, you would have to pad the resource to 65538 bytes, so that the scaled-down value would be 32769, which scales back up to 65538.

I believe this bug was fixed in Windows 98 but I'm not sure. There is a little program in the SDK called fixres95 that generates the necessary padding.

Other limitations of the 16-bit resource file format you may run into:

- Ordinal resource identifiers may not exceed 32767.

- The total lengths of named resources may not exceed 65535 (where each name counts one byte for each character in the name, plus one). Named resources have been a bad idea since Windows 1.0. They are a convenience that you can easily live without most of the time, and they are significantly more costly, as you can see.

Raymond Chen

Follow