

Why are kernel HANDLES always a multiple of four?

 devblogs.microsoft.com/oldnewthing/20050121-00

January 21, 2005



Raymond Chen

Not very well known is that the bottom two bits of kernel HANDLES are always zero; in other words, their numeric value is always a multiple of 4. Note that this applies only to kernel HANDLES; it does not apply to pseudo-handles or to any other type of handle (USER handles, GDI handles, multimedia handles...) Kernel handles are things you can pass to the CloseHandle function.

The availability of the bottom two bits is buried in the `ntdef.h` header file:

```
//  
// Low order two bits of a handle are ignored by the system and available  
// for use by application code as tag bits. The remaining bits are opaque  
// and used to store a serial number and table index.  
//  
#define OBJ_HANDLE_TAGBITS 0x00000003L
```

That at least the bottom bit of kernel HANDLES is always zero is implied by the GetQueuedCompletionStatus function, which indicates that you can set the bottom bit of the event handle to suppress completion port notification. In order for this to work, the bottom bit must normally be zero.

This information is not useful for most application writers, which should continue to treat HANDLES as opaque values. The people who would be interested in tag bits are those who are implementing low-level class libraries or are wrapping kernel objects inside a larger framework.

Raymond Chen

Follow

