

# The parameters to `PostQueuedCompletionStatus` are not interpreted

[devblogs.microsoft.com/oldnewthing/20070525-00](http://devblogs.microsoft.com/oldnewthing/20070525-00)

May 25, 2007



Raymond Chen

Larry Osterman mentioned this almost in passing quite a while ago, that the parameters to `PostQueuedCompletionStatus` are not interpreted by the operating system. Well, obviously the first parameter, the handle to the completion port, is interpreted as the handle to the completion port. But the other parameters, `dwNumberOfBytesTransferred`, `dwCompletionKey`, and `lpOverlapped` are not interpreted at all. Even though they have names, the names don't mean anything. Whatever values you pass for those three parameters merely pop out of `GetQueuedCompletionStatus` when the notification packet makes it to the head of the queue. Why do the parameters have names if the names don't mean anything? Because the operating system itself will post notifications to the completion port if you ask it to, and in that case, the values returned by the `GetQueuedCompletionStatus` function really mean something related to their names. For example, if you issued an asynchronous read to a file that is associated with a completion port, then the `dwNumberOfBytesTransferred` really is the number of bytes transferred, the `dwCompletionKey` really is the completion key that you associated with the file handle when you called `CreateIoCompletionPort`, and the `lpOverlapped` really is the pointer to the `OVERLAPPED` structure that you originally issued the read request against. But if you call `PostQueuedCompletionStatus` manually, then you can choose any values you want. Of course, if you're going to mix operating system-generated completion port activity with manually-generated completion port activity, it would be in your best interest to use the `dwCompletionKey` (or if you're really clever, the `lpOverlapped`) in a consistent manner so that you can tell whether a notification came from the I/O subsystem or was one you generated manually.

But it's up to you.

[Raymond Chen](#)

**Follow**

