If I attach a file to an existing completion port, do I have to close the completion port handle a second time?

devblogs.microsoft.com/oldnewthing/20130823-00

August 23, 2013

Raymond Chen

There are two ways of calling the `CreateIOCompletionPort` function. You can pass a null pointer as the `ExistingCompletionPort` parameter, indicating that you would like to create a brand new completion port, associated with the file handle you passed (if you passed one). Or you can pass the handle of an existing completion port, and the file handle you passed will be associated with that port, in addition to whatever other file handles are already associated with that port.

In both cases, the return value on success is a handle to the I/O completion port, either the brand new one or the existing one. The question from a customer was, “In the case where I am associating a file handle to an existing completion port, do I need to close the completion port handle a second time?” In other words, is there a handle leak in this code:

```
BOOL CCompletionPort::Attach(HANDLE h, ULONG_PTR key)
{
 // you are required to have created the completion port first
 assert(m_hPort != nullptr);
 HANDLE hPort = CreateIoCompletionPort(h, m_hPort, key, 0);
 // line missing? if (hPort != nullptr) CloseHandle(hPort);
 return hPort != nullptr;
}
```

No, there is no handle leak. If you ask to associate a file with an existing completion port, the same handle you passed as the `ExistingCompletionPort` parameter is returned back on success. There is no new obligation to close the handle a second time because kernel handles are not reference-counted. Closing a kernel handle twice is always an error, because the first close closes the handle, and the second close attempts to use a handle that is already closed. There is no reference count on handles.[1]

[1] There is a reference count on kernel objects, but handles are not reference-counted. The standard way to increment the reference count on a kernel object is… to create a new handle to it!

Raymond Chen

Follow