

# When does a process ID become available for reuse?

 [devblogs.microsoft.com/oldnewthing/20110107-00](http://devblogs.microsoft.com/oldnewthing/20110107-00)

January 7, 2011



Raymond Chen

A customer wanted some information about process IDs:

I'm writing some code that depends on process IDs and I'd like to understand better problem of process ID reuse.

When can PIDs be reused? Does it happen when the process handle becomes signaled (but before the zombie object is removed from the system) or does it happen only after last handle to process is released (and the process object is removed from the system)?

If its the former, will `OpenProcess()` succeed for a zombie process? (i.e. the one that has been terminated, but not yet removed from the system)?

The process ID is a value associated with the process object, and as long as the process object is still around, so too will its process ID. The process object remains as long as the process is still running (the process implicitly retains a reference to itself) or as long as somebody still has a handle to the process object. If you think about it, this makes sense, because as long as there is still a handle to the process, somebody can call `WaitForSingleObject` to wait for the process to exit, or they can call `GetExitCodeProcess` to retrieve the exit code, and that exit code has to be stored *somewhere* for later retrieval. When all handles are closed, then the kernel knows that nobody is going to ask whether the process is still running or what its exit code is (because you need a handle to ask those questions). At which point the process object can be destroyed, which in turn destroys the process ID.

What happens if somebody calls `OpenProcess` on a zombie process? The same thing that happens if they call it on a running process: They get a handle to the process. Why would you want to get a handle to a zombie process? Well, you might not know that it's a zombie yet; you're getting the handle so you can call `WaitForSingleObject` to see if it has exited yet. Or you might get the handle, knowing that it's a zombie, because you want to call `GetExitCodeProcess` to see what the exit code was.

Raymond Chen

**Follow**

