

# Additional helpful pseudo-handles: The process token, the thread token, and the effective token

[devblogs.microsoft.com/oldnewthing/20210105-00](https://devblogs.microsoft.com/oldnewthing/20210105-00)

January 5, 2021



Raymond Chen

In addition to the pseudo-handles produced by `GetCurrentProcess()` and `GetCurrentThread()`, there are also pseudo-handles for tokens.

Function	Equivalent
<code>GetCurrentProcessToken</code>	<code>OpenProcessToken(GetCurrentProcess())</code>
<code>GetCurrentThreadToken</code>	<code>OpenThreadToken(GetCurrentThread())</code>
<code>GetCurrentThreadEffectiveToken</code>	<code>OpenThreadToken(GetCurrentThread()) + OpenProcessToken(GetCurrentProcess())</code>

These pseudo-handles are handy because they can be obtained quickly and don't need to be closed. You usually use them for quick one-shot queries.

The thread effective token is particularly useful because getting the thread effective token is normally a bit of a hassle. You first try to get the thread token, to see if the thread is impersonating. If that fails, then you fall back to the process token. This is an annoying bit of boilerplate that can be avoided by going straight to `GetCurrentThreadEffectiveToken()`.

We'll look at an example next time.

**Bonus chatter:** These token pseudo-handles are even more pseudo than your regular pseudo-handles. They basically work only for `GetTokenInformation`. You can't use them with `DuplicateHandle`, for example.

[Raymond Chen](#)

**Follow**

