Some call it context, others call it reference data, but whatever it is, it's yours

devblogs.microsoft.com/oldnewthing/20061218-04

December 18, 2006

Raymond Chen

Different functions call it different things. `RegisterWaitForSingleObject` calls it `Context`. `SetWindowSubclass` calls it reference data. `EnumWindows` calls it `lParam`. `CreateThread` just calls it a parameter! But whatever its name is, it means the same thing: It's a value the function doesn't care about. All the function does is hand that value back to you. What the value means is up to you. What if you need to pass more context than a single pointer? What if you want to pass, say, two pointers? Then put the two pointers in a structure and pass a pointer to that structure. When you do this, you introduce lifetime issues, so make sure you have a plan for deciding who is responsible for freeing the memory when it is no longer needed. "Why isn't this documented in MSDN? Otherwise, people who call `CreateThread` won't know that the parameter needs to be a pointer to this structure and that the thread procedure needs to free the memory."

It's not documented in MSDN because MSDN doesn't care. This is entirely a convention within your program. It's your responsibility to ensure that the code that calls `CreateThread` and the thread procedure agree on what the thread parameter means and how it should be managed.

Raymond Chen

Follow