Try to avoid having BOOL function parameters

devblogs.microsoft.com/oldnewthing/20060828-18

August 28, 2006

Raymond Chen

Generally speaking, I believe that you should try to avoid giving functions a boolean parameter (`BOOL` , `bool` , etc.) unless the meaning of that boolean parameter is blatantly obvious. Examples of obvious meaning would be the second parameter to the `EnableWindow` function (`TRUE` obviously means the window is being enabled and `FALSE` means that it's being disabled) and the final parameter to `ShowScrollBar` (`TRUE` obviously means the scroll bar is being shown and `FALSE` means that it's being hidden). In both of these cases, the meaning of the boolean parameter is encoded in the name of the function itself. But for functions like `CreateEvent` , what does that first `BOOL` parameter mean? First, you have to realize that the first `BOOL` parameter controls whether you get an auto-reset event or a manual-reset one. Does `FALSE` create a manual-reset event? Or is that done by passing `TRUE` ? I can never remember and I have to go looking it up each time. That first parameter should have been declared as, say, a `DWORD` or, even better, an `enum` with two legal values, `EVENTTYPE_AUTORESET` and `EVENTTYPE_MANUALRESET` . Even worse is that `CreateEvent` has **two** `BOOL` parameters. Like anybody reading the code later can remember which comes first. And the mystery `bool` s keep coming. Consider, for example, `StreamReader(Stream, bool)`. What does `true` mean? Or `false` ? Heck if I know.

Mind you, this is just my opinion. Others may disagree with me.

Raymond Chen

Follow