

# Const pointers: Logical consequences

 [devblogs.microsoft.com/oldnewthing/20060908-03](http://devblogs.microsoft.com/oldnewthing/20060908-03)

September 8, 2006



Raymond Chen

Consider this follow-up question to the question from last time:

When I call the `PropertySheet` function, can I assume that the `phpage` field of the `PROPSHEETHEADER` structure will not be modified?

If we take a look at the declaration of the `PropertySheet` function, we see that it reads like this:

```
typedef const PROPSHEETHEADERA *LPCPROPSHEETHEADERA;  
typedef const PROPSHEETHEADERW *LPCPROPSHEETHEADERW;  
WINCOMMCTRLAPI INT_PTR WINAPI PropertySheetA(LPCPROPSHEETHEADERA);  
WINCOMMCTRLAPI INT_PTR WINAPI PropertySheetW(LPCPROPSHEETHEADERW);
```

Go past all the function declaration specification goo and look at the parameter list. It's a `const` pointer to a `PROPSHEETHEADER` structure (either ANSI or Unicode, depending on which flavor of the `PropertySheet` function you're calling).

One of the rules for const pointers is that you can read from them but you cannot write to them. Consequently, the `PropertySheet` function is not allowed to modify the `PROPSHEETHEADER` structure. Assuming your code doesn't modify the `PROPSHEETHEADER` yourself, any value on exit from the function will be the same as the value it had on entry.

Raymond Chen

**Follow**

