

# Why does PrintWindow hate CS\_PARENTDC? Because EVERYBODY hates CS\_PARENTDC!

 [devblogs.microsoft.com/oldnewthing/20120604-00](http://devblogs.microsoft.com/oldnewthing/20120604-00)

June 4, 2012



Raymond Chen

Commenter kero wants to know why the PrintWindow function hates CS\_PARENTDC. (And `CS_CLASSDC`, and `CS_OWNDC`.) Because *everybody* hates `CS_PARENTDC`! (And `CS_CLASSDC`, and `CS_OWNDC`.) We saw earlier that these class styles violate widely-held assumptions about how drawing works. I mean, who would have thought that asking for two device contexts would give you the same one back twice? Or that changes to one device context would secretly modify another (because they're really the same)? Or that a window procedure assumes that it will see only one device context *ever*? The `PrintWindow` function is really in a pickle when faced with a window with one of these class styles, because the whole point of `PrintWindow` is to render into some *other* device context. The `PrintWindow` function says "Render into this other device context, please," and the window acts like a bratty two-year-old who refuses to eat food from anything other than his favorite monkey plate. "This is not my monkey plate. I will now throw a tantrum." The `PrintWindow` function passes a custom device context as a parameter to the `WM_PRINT` message, and the window says, "Hey, no fair! My class styles say that you aren't allowed to pass any old device context; you have to pass *mine*. I will now take my broccoli and mash it all over the table." Oh well. At least it tried.

Yet another reason to avoid these weird class styles.

Raymond Chen

**Follow**

