

# Why do property sheets sometimes take a first-chance exception?

[devblogs.microsoft.com/oldnewthing/20070108-02](http://devblogs.microsoft.com/oldnewthing/20070108-02)

January 8, 2007



Raymond Chen

Reader cmonachan asked [why CPropertySheet::DoModal can cause a first-chance exception](#). This is mentioned in [Knowledge Base article 158552](#). But why take the exception in the first place? First off, let's take MFC out of the picture. The first-chance exception is coming from the property sheet manager. MFC is just the middle-man. Okay, so why the first-chance exception? It's not obviously backwards compatibility, since property sheets were new for Windows 95; there was no old property sheet manager to be compatible with. But yes, it was for compatibility. Not with an imaginary "earlier version" of the property sheet manager, but with real honest-to-goodness dialog box editing tools. Property sheets are [nested dialogs](#). The styles for nested dialogs must include `DS_CONTROL` and `WS_CHILD`. What's more, it can't contain styles like `DS_ABSALIGN` and `WS_CAPTION` which make no sense for child windows. Most of those styles you can convince your dialog box editing tool to add or remove, but `DS_CONTROL` is a different story. Since this flag was new for Windows 95, no existing dialog box editing tool supported it. Okay, so now you're designing a property sheet manager. What do you do with dialog box templates that have the wrong style? **Proposal 1:** Don't write any special code. Just create the dialog box on the assumption that the caller set the styles correctly. The upside of this proposal is that it's the least amount of work. The downside is that if the caller messes up, they just get bizarre behavior (property sheet pages that pop out of the frame, for example) and have no real clue what they did wrong, much less how they can go about fixing it. **Proposal 2:** Write special code to detect incorrect dialog boxes and reject them. Okay, well instead of bizarre behavior, the caller just gets nothing. Still not much help. There's a serious problem with the first two proposals: How do you get people to set the styles correctly if their tools aren't capable of doing it? Fire up the dialog box editing tool you've been using and it won't have a check-box for `DS_CONTROL` because your dialog box editing tool was written for Windows 3.1, and `DS_CONTROL` didn't exist then. You might say, "Well, those people will have to upgrade their dialog box editing tools in order to write programs for Windows 95." The problem with that position is that it creates a chicken-and-egg problem. It's 1994. Windows 95 Beta 1 has just come out. A company wants to "catch the wave" and have a Windows 95 version of their program ready on the day that Windows 95 hits the shelves, so they anxiously install the beta, install the beta SDK, they see this cool new property sheet thing so they sit down to give

it a shot and... they're stuck. They can't use property sheets because they need the Windows 95 version of Microsoft Visual C++ or Delphi or Turbo Pascal. But those products don't exist yet because Windows 95 isn't out yet. And it's not just the development tools that would need to be upgraded. It's also the translation tools and all the other tools that manipulate dialog box templates. This isn't an imaginary scenario. The Windows 95 team faced this very problem! In order to use these cool property sheet things, they needed to add the `DS_CONTROL` style to their dialog boxes, but Visual C++ didn't support this style since that style existed only in an operating system that hadn't been released yet. **Proposal 3:** Write special code to detect incorrect dialog boxes and fix them so that people can start writing Windows 95 programs *now*. To fix the dialog box requires modifying the styles, which means modifying the dialog template, and that's where the first-chance exception comes from. When the property sheet manager writes to the dialog template to fix the style, a first-chance exception is raised because resources start out as read-only pages. The kernel catches this exception and write-enables the page, then returns `EXCEPTION_CONTINUE_EXECUTION` to say, "I fixed the problem, try it again." That's where the first-chance exception comes from.

How do you avoid this first-chance exception? Easy. Get your dialog box styles right in the first place. If you get the styles right, then the property sheet manager won't have to fix them. For the record, here are the style requirements for property sheets:

Must have	Optional
<code>DS_3DLOOK</code> <code>DS_CONTROL</code> <code>WS_CHILD</code> <code>WS_TABSTOP</code>	<code>DS_SHELLFONT</code> <code>DS_LOCALEEDIT</code> <code>WS_CLIPCHILDREN</code>

Any other `WS_*` and `DS_*` styles not listed above are forbidden. (Note that I'm talking about styles and not extended styles.)

Raymond Chen

**Follow**

