Why do I sometimes get classic buttons and sometimes themed buttons depending on the host process?

devblogs.microsoft.com/oldnewthing/20121220-00

December 20, 2012

Raymond Chen

A customer reported that their printer configuration property sheet page looked different depending on the host process. In some processes, the printer configuration dialog had the classic look of Windows 2000, but in other processes it has the themed look of Windows XP and later versions.

> The printer driver calls the `CreatePropertySheetPageW` to create the property sheet page that contains the push-button, radio-button, and other controls. We've confirmed that we call the function with the same parameters each time, but the results are different depending on which program is doing the printing. We've confirmed that both applications are using version 6.0 of the common controls library. Can you provide pointers for investigating why we are getting the old-school look sometimes?

It's clear that the problem is that the property sheet page is being created with a different activation context depending on the host process. If the activation context is a v6 context, then you get the themed buttons; if the activation context is a v5 context, then you get the classic buttons. The fact that version 6.0 of the common controls library is loaded by the process is not relevant; what's relevant is which version is active at the time the dialog is created, since that's the time the class names are resolved. If the problem were with a regular dialog box, then you can explicitly activate the v6 context before calling `DialogBox`, or you can use isolation awareness to have the activation code generated for you. However, the case here is a property sheet page. Since property sheet pages are created on demand (when the user selects the page), you don't have direct control over the code that calls `DialogBox` in order to activate your v6 manifest. Instead, use the `hActCtx` member of the `PROPSHEETPAGE` structure and set the `PSP_USEFUSIONCONTEXT` flag in the `dwFlags`.

Trivia: *Fusion* was the code name for the feature which includes things like isolated applications, application manifests and redirection files. And because they were apparently a bunch of nerds (*quelle surprise*!) they named the initial version *Hydrogen*.

Raymond Chen

Follow