

Why can't I programmatically inspect the check boxes in the Security property sheet any more?

devblogs.microsoft.com/oldnewthing/20221024-00

October 24, 2022



Raymond Chen

A customer reported that they used the `CreateSecurityPage` function to create a security property sheet page. The `ISecurityInformation` object they provide uses the `ISecurityInformation::PropertySheetPageCallback` method to obtain a handle to the property sheet page window, and then they use functions like `IsDlgButtonChecked` to snoop on the check boxes on the page. Maybe they planned on launching fireworks if you set the security options the way they liked. Anyway, they found that this mechanism stopped working in Windows 8 and wanted to know when it would be fixed.

Okay, short answer: It's not going to be fixed, since there is nothing broken in Windows. The broken thing is the customer's program.

The purpose of the `ISecurityInformation::PropertySheetPageCallback` method is to give you some insight into the life cycle of the page. The `PSPCB_CREATE` and `PSPCB_RELEASE` notifications represent the start and end of the page lifetime, and the `PSPCB_SI_INITDIALOG` notification lets you delay initialization until the page becomes active.

The window handle is provided so you have an anchor if you need to display an error message. It's not there so you can reach in and start manipulating the controls on the page (whose control IDs have been and continue to remain undocumented).

In Windows 8, the security page was completely redesigned, and any old code that snooped at the controls on the page will stop working. Which, to be noted, is something you weren't supposed to be doing anyway.

The way to react to the user's actions is to respond to methods like `ISecurityInformation::SetSecurity`, which tell you what the user ultimately settled upon and asks you to apply those changes to the underlying securable object. These methods are how the provider interacts with the page. Snooping around on the page controls is not supported, because those are implementation details and not part of the contract.

Windows 8 refreshed the security UI, and that means a new implementation, so anything that relied on undocumented details of the old implementation will stop working.

Still, this is another example of the difficulty of making any changes to the user interface: There's a good chance that there's somebody who has taken some sort of obscure dependency on how things used to work, and any change you make will cause some third-party program somewhere to break.

Raymond Chen

Follow

