

Just because a method is called Refresh doesn't mean that it refreshes what you want

devblogs.microsoft.com/oldnewthing/20081128-00

November 28, 2008



Raymond Chen

Here's a question from a customer:

I made some changes related to my shell extension [details omitted], but the changes don't show up in the Explorer window when I refresh it. Any suggestions on how to solve this problem?

When we asked how they were refreshing the Explorer window, we were expecting something like pressing F5 or calling `SHChangeNotify` with `SHCNE_UPDATEDIR`, or maybe calling `IShellView::Refresh` or possibly even calling `WebBrowser.Refresh` from script. But we definitely didn't expect this response:

I'm invoking the `Process.Refresh()` method from the `System.Diagnostics` namespace.

Just because a method is called `Refresh` doesn't mean that it refreshes what you want. I think this is somebody who just saw a method name, perhaps inspired by IntelliSense and—**Boom! You have it!**—assumed it did what was needed without actually reading the documentation to check. But you don't even need to read the documentation to know that `Process.Refresh` has no chance of working.

Since it's a method on the `Process` class, the method is applicable to all processes. But certainly there is no generic way to tell a process to refresh. This magical `Refresh` method would have to know how to refresh Explorer windows, Firefox web pages, iTunes mp3 tags... And what would it even mean to refresh, say, a Notepad window? Does that mean to throw away all changes and reload the original document?

How do you know that there is no generic way to tell a process to refresh? Well, for one thing, a single process like Explorer can be viewing multiple objects that can be refreshed; which one are you refreshing? Second, when you write your own program, how do *you* implement refresh? Do you respond to some standard system Refresh message? Or do you just add a

Refresh option to your program's main menu and give it some arbitrary command ID? If there's not even a standard way to refresh *your* program's window, then how can there be a standard way to refresh *all* program windows?

In this specific case, the `Process.Refresh` method refreshes the `Process` object's internal cache of process properties. It doesn't actually do anything to the process itself. How could it?

It's like thinking that the `Matrix.Rotate` method rotates the entries in a matrix.

Epilogue

Actually, I'm scared by this customer's question for another reason: The fact that they even mentioned `Process.Refresh` suggests to me that they wrote their shell extension in managed code, which we already know is strongly disrecommended.

Raymond Chen

Follow

