The curious pattern of pre-emptively rejecting the solution to your problem

devblogs.microsoft.com/oldnewthing/20130206-00

February 6, 2013

Raymond Chen

A frustrating pattern that shows up occasionally in customer questions is the case where the customer poses a problem, and pre-emptively rejects the mechanism explicitly designed to solve that problem.

> How can we change the widget color without using `IWidget::SetColor`?

Um, the whole point of `IWidget::SetColor` is to change the color of a widget. Why are you rejecting the mechanism whose sole purpose in life is to solve the very problem you are having?

Usually, if you press hard enough, they will cough up the reason why they think they cannot use the solution specifically designed to do what they want. Various excuses tend to come up over and over.

One excuse is the belief that the proposed solution does not work in a particular scenario. “We cannot use ACLs because they don’t work on network volumes.” Um, yes they do. Check it out.

Or that the proposed solution doesn’t fit their choice of technology. “We are programming in a language that does not support COM objects. We can only p/invoke to C-style APIs.” Well, you can work around that problem by writing a helper DLL that exposes a C-style API, and implements it by calling the COM method.

Or that the proposed solution violates some vague corporate policy. “We have a corporate policy that users cannot change widget colors, so the `IWidget::SetColor` method returns `E_ACCESSDENIED`. We’re looking for a way around that policy.” Okay, well, now that’s something you need to take up with the people who establish your corporate policies. Don’t come to us looking for ways to circumvent corporate policy.

One time, the reason came from our own technical support staff: “We cannot write a C++ program that calls `IWidget::SetColor` and provide it to the customer because we are not a developer support team. We are not allowed to send compiled binaries to the customer for liability reasons, and we generally do not send source code because our customers typically do not have the expertise or desire to install Visual Studio and the Platform SDK just to compile and run a five-line C++ program. (Did I mention that we are not a developer support team?) Can it be done from a batch file?”

Yeah, how about this batch file:

```
 >changeColor.cs echo using System;
>>changeColor.cs echo class Program {
>>changeColor.cs echo public static void Main(string[] args) {
>>changeColor.cs echo …
>>changeColor.cs echo }
>>changeColor.cs echo }
%windir%\Microsoft.NET\Framework\v4.0.30319\csc changeColor.cs
changeColor blue
```

Only half-joking.

The non-joking answer is “The customer can take this information to a developer support team, or at least somebody who will write the program for them, if they don’t know how to write a program themselves.” Microsoft Consulting Services exists for this, but that is likely overkill for a five-line program.

Raymond Chen

Follow