

# Application compatibility layers are there for the customer, not for the program

[devblogs.microsoft.com/oldnewthing/20100311-00](http://devblogs.microsoft.com/oldnewthing/20100311-00)

March 11, 2010



Raymond Chen

Some time ago, a customer asked this curious question (paraphrased, as always):

Hi, we have a program that was originally designed for Windows XP and Windows Server 2003, but we found that it runs into difficulties on Windows Vista. We've found that if we set the program into Windows XP compatibility mode, then the program runs fine on Windows Vista. What changes do we need to make to our installer so that when the user runs it on Windows Vista, it automatically runs in Windows XP compatibility mode?

Don't touch that knob; the knob is there for the customer, not for the program. And it's there to clean up after your mistakes, not to let you hide behind them.

It's like saying, "I normally toss my garbage on the sidewalk in front of the pet store, and every morning, when they open up, somebody sweeps up the garbage and tosses it into the trash. But the pet store isn't open on Sundays, so on Sundays, the garbage just sits there. How can I get the pet store to open on Sundays, too?"

The correct thing to do is to figure out what your program is doing wrong and fix it. You can use the Application Compatibility Toolkit to see all of the fixes that go into the Windows XP compatibility layer, then apply them one at a time until you find the one that gets your program running again. For example, if you find that your program runs fine once you apply the VersionLie shim, then go and fix your program's operating system version checks.

But don't keep throwing garbage on the street.

[Raymond Chen](#)

**Follow**

