

When setting expectations, you also have to deny them when necessary

 devblogs.microsoft.com/oldnewthing/20100624-00

June 24, 2010



Raymond Chen

Occasionally the shell team will get a request from a customer via their customer liaison that requests information on how to accomplish something or other, and before we'll answer the question, we want to know what the support boundaries are. For example, we might provide a mechanism that works on Windows Vista but comes with no guarantees that it'll work in the future. (This sort of one-off solution might be appropriate for, say, a corporate deployment, where the company controls all the computers in their organization and therefore controls what version of Windows runs on each of them.) The customer liaison responds:

Thanks for the information. I do not plan on giving them any expectations on what to expect in Windows 7. Is this okay?

It's not enough not to give any expectations. You must deny expectations explicitly. If you say nothing, then the customer will assume that the solution will continue to work in the future. Of course, even explicitly denying support in future versions might not be enough. I remember one review of Windows Vista that said that one of its serious flaws was that the Windows XP PowerToys didn't run on it. Even though the Windows XP PowerToys explicitly state right at the top, "PowerToys are for Windows XP only and will not work with Windows Vista." Sometimes the customer fails to realize what it means to rely on behavior that is explicitly not supported in future versions of Windows. I recall more than once, a customer asked for a way to accomplish something, and we provided a mechanism with the explicit warning that it is not guaranteed to work on future versions of Windows. The customer replied, "We understand that this technique may not work on future versions of Windows." I got the impression that customer was not a corporation preparing a Windows deployment but was rather a software vendor developing retail software. I asked a clarifying question: "Do *your customers* understand that this technique may not work on future versions of Windows? Are you going to inform them before installation that the program intentionally relies on behavior which may not work on future versions of Windows? (That way, they can

assess whether they wish to purchase your program.) And are you prepared to support your customers when they upgrade to the next version of Windows and your program stops working?”

This was, apparently, a level of understanding that they had not fully-incorporated. Upon realizing that they were making a decision on behalf of their customers, they decided that perhaps their customers wouldn't be happy with that decision they were about to make.