If you ask a Yes/No question, make sure the user also knows what happens when they say No

devblogs.microsoft.com/oldnewthing/20060911-02

September 11, 2006

Raymond Chen

I was talking with someone last year who had a gripe about a music organizer program. Suppose you create some playlists and then decide, "Oh, nevermind, I don't like this playlist." You highlight the playlist and click "Delete". You then get a dialog box that asks, "Do you want to move the songs in this playlist to the Recycle Bin?" "Well, no, I don't want you to recycle those songs. I want to keep the songs. Just delete the playlist," you say to yourself and you click "No". Unfortunately, the program was asking you, "Do you want to move the songs to the Recycle Bin **or delete the songs permanently from your computer**?" The program had already decided that you wanted to delete the songs themselves when you deleted the playlist. It just wanted to know whether you wanted them gone immediately or just tossed into the Recycle Bin. Fortunately, my friend had backups of the songs that had mistakenly been purged from the computer, but it was still quite shocking to see all the music just plain disappear when there was no expectation that anything of the sort was going to happen. When programs put up Yes/No dialogs, they usually don't have a problem explaining what will happen when you click Yes. But they also have to make sure users understand what will happen when they click No. Window Vista's new Task Dialog makes it easier for programs to make it clearer to users what will happen as the result of pushing a button on a dialog box. Instead of being limited to just "Yes" and "No", you can put more meaningful text on the buttons such as "Save" and "Don't Save", or you could use command buttons and provide an explanatory sentence for each option. Now, programs could always have built custom dialogs with these more descriptive buttons, but doing so meant designing a dialog box from scratch, positioning the buttons precisely according to dialog box layout guidelines, and then writing a custom dialog procedure to handle this new custom dialog. Most people just take the easy way out and use `MessageBox`. In Windows Vista there is now a way to build slightly more complex dialogs without having to design a dialog template.

Be careful, however, not to fall into the same trap with task dialogs. The original dialog might have been converted to a task dialog with the buttons "Recycle" and "Don't Recycle", which would not have solved the problem at all.

Raymond Chen

Follow