How to set focus in a dialog box

devblogs.microsoft.com/oldnewthing/20040802-00

August 2, 2004

Raymond Chen

Setting focus in a dialog box is more than just calling SetFocus.

A dialog box maintains the concept of a “default button” (which is always a pushbutton). The default button is typically drawn with a distinctive look (a heavy outline or a different color) and indicates what action the dialog box will take when you hit Enter. Note that this is **not** the same as the control that has the focus.

For example, open the Run dialog from the Start menu. Observe that the OK button is the default button; it has a different look from the other buttons. But focus is on the edit control. Your typing goes to the edit control, until you hit Enter; the Enter activates the default button, which is OK.

As you tab through the dialog, observe what happens to the default button. When the dialog box moves focus to a pushbutton, that pushbutton becomes the new default button. But when the dialog box moves focus to something that isn’t a pushbutton at all, the OK button resumes its position as the default button.

The dialog manager remebers which control was the default button when the dialog was initially created, and when it moves focus to something that isn’t a button, it restores that original button as the default button.

You can ask a dialog box what the default button is by sending it the DM_GETDEFID message; similarly, you can change it with the DM_SETDEFID message.

(Notice that the return value of the DM_GETDEFID message packs the control ID in the low word and flags in the high word. Another place where expanding dialog control IDs to 32-bit values doesn’t buy you anything.)

As the remarks to the DM_SETDEFID function note, messing directly with the default ID carelessly can lead to odd cases like a dialog box with two default buttons. Fortunately, you rarely need to change the default ID for a dialog.

A bigger problem is using SetFocus to shove focus around a dialog. If you do this, you are going directly to the window manager, bypassing the dialog manager. This means that you can create “impossible” situations like having focus on a pushbutton without that button being the default!

To avoid this problem, don’t use SetFocus to change focus on a dialog. Instead, use the WM_NEXTDLGCTL message.

```
void SetDialogFocus(HWND hdlg, HWND hwndControl)
{
 SendMessage(hdlg, WM_NEXTDLGCTL, (WPARAM)hwndControl, TRUE);
}
```

As the remarks for the WM_NEXTDLGCTL message observe, the DefDlgProc function handles the WM_NEXTDLGCTL message by updating all the internal dialog manager bookkeeping, deciding which button should be default, all that good stuff.

Now you can update dialog boxes like the professionals, avoiding oddities like having no default button, or worse, multiple default buttons!

Raymond Chen

Follow