Returning values from a dialog procedure

devblogs.microsoft.com/oldnewthing/20031107-00

November 7, 2003

Raymond Chen

For some reason, the way values are returned from a dialog procedure confuses people, so I'm going to try to explain it a different way.

The trick with dialog box procedures is realizing that they actually need to return **two** pieces of information:

- Was the message handled?
- If so, what should the return value be?

Since two pieces of information have to be returned, but a C function can have only one return value, there needs to be some other way to return the second piece of information.

The return value of the dialog procedure is whether the message was handled. The second piece of information – what the return value should be – is stashed in the DWLP_MSGRESULT window long.

In other words, DefDlgProc goes something like this:

```
LRESULT CALLBACK DefDlgProc(
    HWND hdlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    DLGPROC dp = (DLGPROC)GetWindowLongPtr(hdlg, DWLP_DLGPROC);
    SetWindowLongPtr(hdlg, DWLP_MSGRESULT, 0);
    BOOL_PTR fResult = dp(hdlg, uMsg, wParam, lParam);
    if (fResult) return GetWindowLongPtr(hdlg, DWLP_MSGRESULT);
    else ... do default behavior ...
}
```

If you return anything other than 0, then the value you set via `SetWindowLongPtr(hdlg, DWLP_MSGRESULT, value)` is used as the message result.

For example, many WM_NOTIFY notifications allow you to override default behavior by returning TRUE. To prevent a listview label from being edited, you can return TRUE from the LVN_BEGINLABELEDIT notification. But if you are doing this from a dialog procedure, you have to do this in two steps:

```
SetWindowLongPtr(hdlg, DWLP_MSGRESULT, TRUE);
return TRUE;
```

The second line sets the return value for the dialog procedure, which tells DefDlgProc that the message has been handled and default handling should be suppressed. The first line tells DefDlgProc what value to return back to the sender of the message (the listview control). If you forget either of these steps, the desired value will not reach the listview control.

Notice that `DefDlgProc` sets the DWLP_MSGRESULT to zero before sending the message. That way, if the dialog procedure neglects to set a message result explicitly, the result will be zero.

This also highlights the importance of calling `SetWindowLongPtr` **immediately** before returning from the dialog procedure and no sooner. If you do anything between setting the return value and returning TRUE, that may trigger a message to be sent to the dialog procedure, which would set the message result back to zero.

Caution: There are a small number of "special messages" which do not follow this rule. The list is given in the documentation for DialogProc. Why do these exceptions exist? Because when the dialog manager was first designed, it was determined that special treatment for these messages would make dialog box procedures easier to write, since you wouldn't have to go through the extra step of setting the DWLP_MSGRESULT. Fortunately, since those original days, nobody has added any new exceptions. The added mental complexity of remembering the exceptions outweigh the mental savings of not having to write one line of code ("SetWindowLongPtr(hdlg, DWLP_MSGRESULT, desiredResult)").

Raymond Chen

Follow