What about BOZOSLIVEHERE and TABTHETEXTOUTFORWIMPS?

devblogs.microsoft.com/oldnewthing/20031015-00

October 15, 2003

Raymond Chen

In a previous entry I discussed the story behind the functions with the funny names BEAR, BUNNY and PIGLET. But what about the ones with goofier names like BOZOSLIVEHERE and TABTHETEXTOUTFORWIMPS?

For this, you need a deeper history lesson.

Back in the old days of real-mode Windows, all callback functions had to be exported. This was necessary for complicated technical reasons I may bother to explain if anybody really cared but I doubt anybody does any more. So the window procedures for all of the standard window classes (edit controls, list boxes, check boxes, etc.) were exported from USER. So too were various other callback functions like timer procedures. This was in addition to the usual collection of internal functions so USER, KERNEL and GDI could coordinate their efforts.

Some people reverse-engineered all these internal functions and printed books on how they worked, and as a result there were a lot of programs that actually used them. Which was quite a surprise to us because they were internal functions. And then when we wanted to redesign these internal functions (for example, to add a parameter, or if we decided that we didn't need it any more and tried to delete it), we found that the programs stopped working.

So we had to put the functions back, with their old behavior. The new features we were contemplating had to be redesigned, redirected, or possibly even abandoned entirely. (If we wanted to delete a function, then the work could continue, but the old function had to stay around with its old behavior. It was basically dead code from the OS's point of view, hanging around just because some random app or other decided to cheat and bypass the documented way of doing things.) But to teach people a lesson, they often got given goofy names.

For example, BOZOSLIVEHERE was originally the window procedure for the edit control, with the rather nondescript name of EditWndProc. Then some people who wanted to use the edit control window procedure decide that GetWindowLong(GWL_WNDPROC) was too much typing, so they linked to EditWndProc directly. Then when Windows 2.0 (I think) removed the need to export window procedures, we removed them all, only to find that programs stopped working. So we had to put them back, but they got goofy names as a way of scolding the programs that were doing these invalid things.

Things got even worse in Windows 95, when all our window procedures were converted to 32-bit versions. The problem is that the old window procedures were only 16-bit. So we couldn't even simply export the 32-bit window procedure under the name BOZOSLIVEHERE. We had to write a conversion function that took an illegal 16-bit function call and converted it to the corresponding illegal 32-bit function call.

This is just the tip of the iceberg with respect to application compatibility. I could probably write for months solely about bad things apps do and what we had to do to get them to work again (often in spite of themselves). Which is why I get particularly furious when people accuse Microsoft of maliciously breaking applications during OS upgrades. If any application failed to run on Windows 95, I took it as a personal failure. I spent many sleepless nights fixing bugs in third-party programs just so they could keep running on Windows 95. (Games were the worst. Often the game vendor didn't even care that their program didn't run on Windows 95!)

Raymond Chen

Follow