How do I prevent users from using the mouse to drag the trackbar thumb to positions that aren't multiples of five? Part 1: Reframe the problem

October 25, 2018

Raymond Chen

A customer had a trackbar common control, set the range to 0–100, set the line size to 5 (so that pressing an arrow key moves the thumb by five units), and set the page size to 20 (so that pressing the `PgUp` and `PgDn` keys move the thumb by twenty units). This ensures that the keyboard can set the trackbar position only to values that are multiples of five. But the user can still drag the thumb with the mouse and set the position to any value, even values that aren't multiples of five. The customer wanted to limit the trackbar to positions that are multiples of five.

The trackbar allows the thumb position to be any value between the minimum and maximum. It's like a scroll bar in that way. If you want the trackbar to operate only in multiples of five, then just divide everything by five. Set the range to 0–20, set the line size to 1, and set the page size to 4. When you read the trackbar position, just multiply it by five when you read the value out.

Take the scratch program and make the following changes:

```
#include <strsafe.h>

BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
 g_hwndChild = CreateWindow(TRACKBAR_CLASS, TEXT(""),
    WS_CHILD | WS_VISIBLE,
    0, 0, 100, 100,
    hwnd, (HMENU)100, g_hinst, 0);

  SendMessage(g_hwndChild, TBM_SETRANGE, TRUE, MAKELPARAM(0, 20));
  SendMessage(g_hwndChild, TBM_SETPAGESIZE, 0, 4);

  return TRUE;
}
```

The line size defaults to 1, so we didn't have to set it explicitly.

```
void OnHScroll(HWND hwnd, HWND hwndCtl, UINT code, int pos)
{
 if (hwndCtl == g_hwndChild) {
  TCHAR buf[128];
  pos = (int)SendMessage(hwndCtl, TBM_GETPOS, 0, 0) * 5;
  StringCchPrintf(buf, ARRAYSIZE(buf), TEXT("pos = %d"), pos);
  SetWindowText(hwnd, buf);
 }
}

    HANDLE_MSG(hwnd, WM_HSCROLL, OnHScroll);
```

When we are told that the scroll position changed, we fetch the current trackbar position, *multiply it by five*, and then show the result.

The user can drag the trackbar thumb to any unit position, but since we multiply the result by five, the effective behavior is that they can drag the thumb to any position that is a multiple of five.

This works great until you get a bug report (perhaps from a government agency) that your program is not accessible because the accessibility tools report a scroll bar whose minimum, maximum, and position are all off by a factor of five.

To fix this problem, we'll have to take a different approach. We'll take up this issue next time.

Raymond Chen

Follow