EnumChildWindows already enumerates recursively

devblogs.microsoft.com/oldnewthing/20070116-04

January 16, 2007

Raymond Chen

I often see people write code that goes something like this:

```
// do not use - see discussion
void DoSomethingToWindowTree(HWND hwndRoot)
{
 // first do it to the window passed in
 DoSomething(hwndRoot);
 // now do it to all the children
 EnumChildWindows(hwndRoot, DoSomethingHelper, 0);
}
BOOL CALLBACK DoSomethingHelper(HWND hwnd, LPARAM lParam)
{
 DoSomethingToWindowTree(hwnd);
 return TRUE;
}
```

The intent here was to perform the operation on all the windows in a window tree by operating on the root, then operating on each of the children. Operating on the children is in turn performed recursively, so that we eventually see every window in the tree.

Except that if you actually run this function on a vaguely interesting window tree, you'll find that items get counted multiple times.

The reason is that the `EnumChildWindows` function **already does the recursion**:

> If a child window has created child windows of its own, **EnumChildWindows** enumerates those windows as well.

If you add your own recursion, then you end up counting grandchildren twice, great-grandchildren four times, and so on. The recursion is already done by `EnumChildWindows`; just use it.

```
void DoSomethingToWindowTree(HWND hwndRoot)
{
 // first do it to the window passed in
 DoSomething(hwndRoot);
 // now do it to all the descendants (children, grandchildren, etc.)
 EnumChildWindows(hwndRoot, DoSomethingHelper, 0);
}
BOOL CALLBACK DoSomethingHelper(HWND hwnd, LPARAM lParam)
{
 DoSomething(hwnd);
 return TRUE;
}
```

Raymond Chen

Follow