

We've traced the call and it's coming from inside the house: A function call that always fails

 devblogs.microsoft.com/oldnewthing/20111103-00

November 3, 2011



Raymond Chen

A customer reported that they had a problem with a particular function added in Windows 7. The tricky bit was that the function was used only on very high-end hardware, not the sort of thing your average developer has lying around.

```
GROUP_AFFINITY GroupAffinity;
... code that initializes the GroupAffinity structure ...
if (!SetThreadGroupAffinity(hThread, &GrouAffinity, NULL));
{
    printf("SetThreadGroupAffinity failed: %d\n", GetLastError());
    return FALSE;
}
```

The customer reported that the function always failed with error 122 (`ERROR_INSUFFICIENT_BUFFER`) even though the buffer seems perfectly valid.

Since most of us don't have machines with more than 64 processors, we couldn't run the code on our own machines to see what happens. People asked some clarifying questions, like whether this code is compiled 32-bit or 64-bit (thinking that maybe there is an issue with the emulation layer), until somebody noticed that there was a stray semicolon at the end of the `if` statement.

The customer was naturally embarrassed, but was gracious enough to admit that, yup, removing the semicolon fixed the problem.

This reminds me of an incident many years ago. I was having a horrible time debugging a simple loop. It looked like the compiler was on drugs and was simply ignoring my loop conditions and always dropping out of the loop. At wit's end, I asked a colleague to come to my office and serve as a second set of eyes. I talked him through the code as I single-stepped:

“Okay, so we set up the loop here...”

```
NODE pn = GetActiveNode();
```

“And we enter the loop, continuing while the node still needs processing.”

```
if (pn->NeedsProcessing())  
{
```

“Okay, we entered the loop. Now we realign the skew rods on the node.”

```
pn->RealignSkewRods();
```

“If the treadle is splayed, we need to calibrate the node against it.”

```
if (IsSplayed()) pn->Recalibrate(this);
```

“And then we loop back to see if there is more work to be done on this node.”

```
}
```

“But look, even though the node needs processing «view node members», we don’t loop back. We just drop out of the loop. What’s going on?”

— *Um, that’s an `if` statement up there, not a `while` statement.*

A moment of silence while I process this piece of information.

“All right then, sorry to bother you, hey, how about that sporting event last night, huh?”

Raymond Chen

Follow

