
Subject: Re: Problems with Dave's LinkedList
Posted by [davidf](#) on Mon, 02 Apr 2001 18:55:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

ALAN FRAZIER (s007amf@news.wright.edu) writes:

```
> I am using Dave Fanning's LinkedList object and seem to have run into a
> problem with it. In the procedure Delete_Node, I was getting a Invalid
> Pointer error on line:
>
> IF NOT Ptr_Valid((*currentNode).previous) THEN BEGIN
>
> To me, it looks like the error occurs when the linked list only has one
> node left. Dave has code that should deal with this condition on line:
>
> IF index EQ (self.count - 1) THEN self->Delete_Last_Node
>
> This code does deal with the above condition, but does not cause
> Delete_Node to exit. Therefore, the list is empty and tries to execute
> the first line I mentioned; which, causes a runtime error.
>
> I changed the second line I mentioned into the following block of code
> (which seems to fix the error):
>
>
> IF index EQ (self.count - 1) THEN BEGIN
>   self->Delete_Last_Node
>   RETURN
> END
```

Ah, this is a problem, I think, in how programs sometimes grow organically.

This was one of the first programs I wrote shortly after pointers were introduced in IDL. I wanted to learn how to use them, and I thought a linked list was a good programming exercise. It was that, and much, much more. When I was finished I pretty much didn't want to *look* at another pointer for a long, long time!

It is a mystery to me why something so simple in theory can be so exasperatingly complex in detail. I grimace just thinking about the experience.

But, anyway, I set out to do one thing, and I ended up doing something else. And by the time I was done, the program worked, but it was a lot more fragile than I expected it to be. When I decided to publish it, I knew

it had to be more robust, but the thought of re-writing all that tedious code was still too fresh.

So what I did instead was add a robust public interface to the private methods I had already worked out. I kind of liked this solution because it tied in nicely with a little lecture I was giving about pointers and objects. But with two years hindsight, I can see it may have not been the brightest idea I ever had. But, in any case, I thought I documented the public methods pretty well.

The point of this long-winded story is that you are using one of the private methods (DELETE_NODE) instead of the proper public method (DELETE). The public method will call the appropriate private methods, and will do so appropriately and without having to resort to making changes in the code. (Thank God!!)

Cheers,

David

P.S. Let's just say I can see the benefit of re-writing this program, but, unfortunately, it is WAY down on my list. :-)

--

David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155
