Subject: IDL 5.5 "% Temporary variables" messages
Posted by Liam E. Gumley on Fri, 24 May 2002 20:22:04 GMT

View Forum Message <> Reply to Message

I just started using IDL 5.5 on SGI/IRIX for everyday tasks, and I've started to see

% Temporary variables are still checked out - cleaning up...

messages that I haven't seen since the IDL 5.2 days (I've been using IDL 5.3 for a couple of years now). Has anyone else seen these messages in IDL 5.5?

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: IDL 5.5 "% Temporary variables" messages
Posted by muswick on Sat, 01 Jun 2002 08:30:36 GMT

View Forum Message <> Reply to Message

I still get them all the time even IDL 5.5 under windows. In mycase, they are caused by entirely by poorly written IDL code - no external calls etc.

In IDL you can access the internals (structure elements) of objects that are in use by dereferrencing a pointer that points to the object. If you have nested objects and pointers to arrays within those objects, attempting to dereference on a single line forces IDL to create internal temporary variables for each dereference while parsing the line. When the value of the line is returned, IDL attempts to clean up those temporary variables it needed. What I believe happens is that, when items are nested on a single line, an early top level temporary variable is destroyed that is itself a heap variable. Thus IDL is left with a dangling pointer. It recognizes the dangling pointer during garbage collection and returns the error.

I got around the problem by breaking up the lines using my own temporary pointers, object references, etc. Then I can destroy them properly.

My first line though indicates the real problem - poorly written IDL code. By accesing the internal structures of an object outside of the object code is violation of data/method hiding that object programming provides.

New code that I write either uses and accessor method, or I have
determined that what I wanted to do, is better performed elsewhere.

Below are two such statements that have caused IDL to return

```
  ((*tvol.window_objs)[i]).zoom = dzoom

    FOR j = 0, N_ELEMENTS(*((*tvol.window_objs)[i]).drawobj)-1 DO
BEGIN
        ((*((*tvol.window_objs)[i]).drawobj)[j])->Display_Slice, $
        VIEW = ((*((*tvol.window_objs)[i]).drawobj)[j]).view
    ENDFOR
```

As you can see, I am accessing the internal structure elements of
various objects.

The work-around for the first statement is:

```
  tmp_obj = (*tvol.window_objs)[i]  ; set the temporary object
reference
  tmp_obj.zoom = dzoom
```

The code worked fine, but I got tired of getting those
% Temporary variables are still checked out - cleaning up...
messages. The work-around gets rid of the massages, but better coding
prevents it all together.

Gary Muswick
University Hospitals of Cleveland

"Liam E. Gumley" <Liam.Gumley@ssec.wisc.edu> wrote in message
news:<3CEEA0EC.50765EE7@ssec.wisc.edu>...

> I just started using IDL 5.5 on SGI/IRIX for everyday tasks, and I've
> started to see
>
> % Temporary variables are still checked out - cleaning up...
>
> messages that I haven't seen since the IDL 5.2 days (I've been using IDL
> 5.3 for a couple of years now). Has anyone else seen these messages in
> IDL 5.5?
>
> Cheers,
> Liam.
> Practical IDL Programming

> http://www.gumley.com/