Subject: How many selfs do I have?
Posted by wlandsman on Thu, 30 Jan 2014 18:58:16 GMT
View Forum Message <> Reply to Message

I was stepping through an object method using the debugger.   As I reached the end of the method I was taken to the object cleanup method, with lines such as the following:

```
 if PTR_VALID(self.pPts) then ptr_free, self.pPts
 if OBJ_VALID(self.oPolyLine) then obj_destroy, self.oPolyLine
 if OBJ_VALID(self.oPolyGon) then obj_destroy, self.oPolyGon
```

At the end of the cleanup method, I was taken back to the start of the cleanup method.   So I stepped through the cleanup again -- and was again taken back to the start.   Ultimately the cleanup method was called four times.   Each time through, if I did a "help, self" I would get a different ID, e.g.

```
SELF            OBJREF    = <ObjHeapVar48371(FIELDPOINTGROUP)>
SELF            OBJREF    = <ObjHeapVar48445(FIELDPOINTGROUP)>
SELF            OBJREF    = <ObjHeapVar48524(FIELDPOINTGROUP)>
SELF            OBJREF    = <ObjHeapVar48551(FIELDPOINTGROUP)>
```

Does anyone have any idea what might be going on here, or what further diagnostics I might use?   Even if I have multiple instances of the object (which I am not aware of) do they all get destroyed even when I am only using one instance?

This part of a very large widget application which typically crashes IDL (dumping core) about once a day.   So I am trying to look for ways to clean the code, e.g. to remove unnecessary object creation.

Thanks, --Wayne

```
print,!Version
{ x86_64 darwin unix Mac OS X 8.3 Nov 15 2013      64      64}
```

Subject: Re: How many selfs do I have?
Posted by David Fanning on Thu, 30 Jan 2014 19:08:27 GMT
View Forum Message <> Reply to Message

wlandsman writes:

> I was stepping through an object method using the debugger.   As I reached the end of the method I was taken to the object cleanup method, with lines such as the following:
>
>   if PTR_VALID(self.pPts) then ptr_free, self.pPts
>   if OBJ_VALID(self.oPolyLine) then obj_destroy, self.oPolyLine
>   if OBJ_VALID(self.oPolyGon) then obj_destroy, self.oPolyGon

>
> At the end of the cleanup method, I was taken back to the start of the cleanup method. So I stepped through the cleanup again -- and was again taken back to the start. Ultimately the cleanup method was called four times. Each time through, if I did a "help, self" I would get a different ID, e.g.
>

```
>  SELF           OBJREF    = <ObjHeapVar48371(FIELDPOINTGROUP)>
>  SELF           OBJREF    = <ObjHeapVar48445(FIELDPOINTGROUP)>
>  SELF           OBJREF    = <ObjHeapVar48524(FIELDPOINTGROUP)>
>  SELF           OBJREF    = <ObjHeapVar48551(FIELDPOINTGROUP)>
```

>
> Does anyone have any idea what might be going on here, or what further diagnostics I might use? Even if I have multiple instances of the object (which I am not aware of) do they all get destroyed even when I am only using one instance?

I would say this is evidence that you have four instances of this object class in your program, since each object has a unique identifier, and objects get unique identifiers only when they are created. If you think you should only have one, then I would try putting a breakpoint in the INIT method and see if you can find out (maybe Help, /Traceback) where each of them is being created.

> This part of a very large widget application which typically crashes IDL (dumping core) about once a day. So I am trying to look for ways to clean the code, e.g. to remove unnecessary object creation.

Object programming can definitely get complicated. I've had object programs that drew their graphics several times in succession, even though I was expecting just once. Part of being "smart" is doing things autonomously. But, it's kind of like having smart children. Better in theory than in practice. ;-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")