
Subject: Re: Inheritance query

Posted by [davidf](#) on Thu, 04 Nov 1999 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Bernard Puc (bpuc@va.aetc.com) writes:

- > For the object programming gurus: I'm creating a class called data.
- > I'm then creating subclasses of data called type1, type2, etc. The
- > type1 class inherits the data class attributes. Now, is it possible to
- > inherit, lets say, the data::INIT method and somehow add to it? Or, do
- > I have to write an entirely new INIT method for type1 class which
- > incorporates the statements in the data::INIT method?

If you don't define an INIT method for, say, your TYPE1 object, IDL will look for an INIT method in a subclass object and use that. If you decide that you would like to add to an INIT method for the TYPE1 object, and still use the INIT method of a superclass, you can call the superclass INIT method from within the TYPE1 INIT method. (The only time you can call a lifecycle method yourself is from within a subclass lifecycle method.) It might look something like this:

```
FUNCTION TYPE1::INIT, _Extra=extra
  ok = self->DATA::INIT(_Extra=extra)
  IF NOT ok THEN RETURN, 0
```

Note this really strange behavior: If you don't create an INIT method for your TYPE1 object and then create a TYPE1 object, it will, of course, use the INIT method of the inherited DATA object. But if you now decide to write a TYPE1::INIT method, you will never be able to attach it to a TYPE1 object in that IDL session! You will have to exit IDL and restart for the correct INIT method to be associated with the TYPE1 object.

I'm just about done with an article explaining this phenomemon in more detail. It should be up on my web page later today.

Cheers,

David

--

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

Subject: Re: Inheritance query
Posted by [davidf](#) on Fri, 05 Nov 1999 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Struan Gray (struan.gray@sljus.lu.se) writes:

> I'm confused. I got the point that if you don't have any INIT or
> CLEANUP method when the first instance of an object is created, it is
> impossible to add one later in the same session, but your article
> seems to imply that it is impossible to modify those methods either,
> even if they successfully compile and run the first time round. I do
> the latter all the time; in fact, it's one of the joys of non-blocking
> widgets. The class *structure* can't be edited, but the lifecycle
> methods can (IDL 5.2.1 on MacOS if it matters).

Yes. I ran out of time yesterday and put that article up before I had tested everything one more time. I was writing it from sketchy notes I had made in recent IDL courses. I'll do more careful editing today.

But I think you are right that the INIT and CLEANUP methods can be edited once they have been properly associated with the object. Frankly, in classes, once we get the damn things to work we don't touch them anymore. :-)

> If the INIT or CLEANUP methods crash so that I get bounced back to
> the command line with the offending routine opened for editing, IDL
> seems to be in a very odd state, but typing RETALL and HEAP_GC several
> times (at least twice) gets me back to the point where I can
> successfully edit and recompile.

Ah, of course. Now I see why this feature is undocumented. :-)

I think I'll stick with my suggestion to exit IDL.
At least I can come up with a believable story for *that* recommendation.

> A final tip: it is worth including an INIT and CLEANUP routine for
> all top-level classes (ie, those that don't inherit anything). If you

- > don't, IDL rummages around in your currently-defined search path
- > looking for one every time you create or destroy an instance of that
- > class, which can severely degrade performance.

Yes, I meant to include this caveat and just completely forgot.
Thanks for reminding me. I'll do it now.

Cheers,

David

--

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

Subject: Re: Inheritance query
Posted by [Struan Gray](#) on Fri, 05 Nov 1999 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning, davidf@dfanning.com writes:

- > I'm just about done with an article explaining
- > this phenomemon in more detail. It should be up
- > on my web page later today.

I'm confused. I got the point that if you don't have any INIT or CLEANUP method when the first instance of an object is created, it is impossible to add one later in the same session, but your article seems to imply that it is impossible to modify those methods either, even if they successfully compile and run the first time round. I do the latter all the time; in fact, it's one of the joys of non-blocking widgets. The class *structure* can't be edited, but the lifecycle methods can (IDL 5.2.1 on MacOS if it matters).

If the INIT or CLEANUP methods crash so that I get bounced back to the command line with the offending routine opened for editing, IDL seems to be in a very odd state, but typing RETALL and HEAP_GC several times (at least twice) gets me back to the point where I can successfully edit and recompile.

A final tip: it is worth including an INIT and CLEANUP routine for all top-level classes (ie, those that don't inherit anything). If you don't, IDL rummages around in your currently-defined search path looking for one every time you create or destroy an instance of that class, which can severely degrade performance.

Struan
