Larry Kneller wrote:
>   Hi All,
>
> I found this "bug" that appears in some of my code now, and describing
> it may be a little confusing.
> In what follows I refer to a function called "class_name()" that I use
> to create an instance of the class
> "class_name" that is defined in class_name__define.pro.
>
> I have a lot of object compound widgets where I make a function that
> (for clarity) has the same name as my compound widget class, and it
> returns the widget_id:
>
> id = class_name(obj=obj, etc)
>
> Here I use the keyword "obj" to get the instance of my class called
> "class_name" that is created
> in the obj_new('class_name',...) call in the widget creation function
> called "class_name".
>
> Now, in IDL8 when I call the widget_creation function "class_name",
>
> id=class_name(obj=obj,etc.)
>
> this is a direct call to "class_name::init" method in my
> class_name__define program, and if
> the call does not throw an error, instead of returning a widget id it
> returns and object reference!!!

Is this due to the IDL8 introduction of

  x = class_name(...)

being equivalent to

  x = OBJ_NEW("class_name",...)

?

Not having yet gotten IDL8 installed yet I haven't been able to check that the above was definitely extended to
user-defined classes. Previous threads have discussed it for the new list() object (i.e. people with user-defined
functions called "list" ran into problems IIRC).

cheers,

paulv

Subject: Re: IDL8 CWO issue
Posted by Michael Galloy on Tue, 07 Sep 2010 16:19:03 GMT
View Forum Message <> Reply to Message

On 9/7/10 9:43 AM, Paul van Delst wrote:
> Larry Kneller wrote:
>>   Hi All,
>>
>> I found this "bug" that appears in some of my code now, and describing
>> it may be a little confusing.
>> In what follows I refer to a function called "class_name()" that I use
>> to create an instance of the class
>> "class_name" that is defined in class_name__define.pro.
>>
>> I have a lot of object compound widgets where I make a function that
>> (for clarity) has the same name as my compound widget class, and it
>> returns the widget_id:
>>
>> id = class_name(obj=obj, etc)
>>
>> Here I use the keyword "obj" to get the instance of my class called
>> "class_name" that is created
>> in the obj_new('class_name',...) call in the widget creation function
>> called "class_name".
>>
>> Now, in IDL8 when I call the widget_creation function "class_name",
>>
>> id=class_name(obj=obj,etc.)
>>
>> this is a direct call to "class_name::init" method in my
>> class_name__define program, and if
>> the call does not throw an error, instead of returning a widget id it
>> returns and object reference!!!
>
> Is this due to the IDL8 introduction of
>
>   x = class_name(...)
>
> being equivalent to
>
>   x = OBJ_NEW("class_name",...)

>
> ?
>
> Not having yet gotten IDL8 installed yet I haven't been able to check that the above was definitely extended to
> user-defined classes. Previous threads have discussed it for the new list() object (i.e. people with user-defined
> functions called "list" ran into problems IIRC).

Yes, this definitely works for user-defined classes also.

Mike
--
www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: IDL8 CWO issue
Posted by Michael Galloy on Tue, 07 Sep 2010 16:21:09 GMT
View Forum Message <> Reply to Message

On 9/7/10 8:10 AM, Larry Kneller wrote:
>   Hi All,
>
> I found this "bug" that appears in some of my code now, and describing
> it may be a little confusing.
> In what follows I refer to a function called "class_name()" that I use
> to create an instance of the class
> "class_name" that is defined in class_name__define.pro.
>
> I have a lot of object compound widgets where I make a function that
> (for clarity) has the same name as my compound widget class, and it
> returns the widget_id:
>
> id = class_name(obj=obj, etc)
>
> Here I use the keyword "obj" to get the instance of my class called
> "class_name" that is created
> in the obj_new('class_name',...) call in the widget creation function
> called "class_name".
>
> Now, in IDL8 when I call the widget_creation function "class_name",
>
> id=class_name(obj=obj,etc.)
>
> this is a direct call to "class_name::init" method in my
> class_name__define program, and if

> the call does not throw an error, instead of returning a widget id it
> returns and object reference!!!
>
> For now, my solution to deal with this is to keep the name of the
> widget creation function the same and
> change the name of the class to "class_name_class"
> so that my calls to the widget creation function (that are peppered
> throughout my applications)
> are left untouched and I can fix this in one place per incident.

It sounds like you have a workaround for this already, but in case you
are still looking for suggestions I think I would add an `identifier`
property::

  cwObject = class_name(identifier=id, ...)

Mike
--
www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: IDL8 CWO issue
Posted by Paul Adams on Tue, 07 Sep 2010 16:25:24 GMT
View Forum Message <> Reply to Message

On Sep 7, 8:10 am, Larry Kneller <larry.knel...@gmail.com> wrote:
> Hi All,
>
> I found this "bug" that appears in some of my code now, and describing
> it may be a little confusing.
> In what follows I refer to a function called "class_name()" that I use
> to create an instance of the class
> "class_name" that is defined in class_name__define.pro.
>
> I have a lot of object compound widgets where I make a function that
> (for clarity) has the same name as my compound widget class, and it
> returns the widget_id:
>
> id = class_name(obj=obj, etc)
>
> Here I use the keyword "obj" to get the instance of my class called
> "class_name" that is created
> in the obj_new('class_name',...) call in the widget creation function
> called "class_name".
>
> Now, in IDL8 when I call the widget_creation function "class_name",

>
> id=class_name(obj=obj,etc.)
>
> this is a direct call to "class_name::init" method in my
> class_name__define program, and if
> the call does not throw an error, instead of returning a widget id it
> returns and object reference!!!
>
> For now, my solution to deal with this is to keep the name of the
> widget creation function the same and
> change the name of the class to "class_name_class"
> so that my calls to the widget creation function (that are peppered
> throughout my applications)
> are left untouched and I can fix this in one place per incident.
>
> -Larry

Hi Larry,

It is the case that in IDL 8.0, the statements "o = class_name(...)"
and "o = OBJ_NEW("class_name",...)" are equivalent. However, in the
case where you have a function named "class_name" and a class named
"class_name", the function should take priority, so the behavior you
are describing is unexpected.

Is your function "class_name" defined in its own file,
"class_name.pro", and is it on your IDL_PATH?

Would it be possible for you to provide an example of failing code?

Thanks in advance,

Paul Adams
ITT VIS