Subject: Re: Treating an object as a structure
Posted by David Fanning on Wed, 04 Mar 2009 20:40:42 GMT
View Forum Message <> Reply to Message

llo writes:

> If I've a struct I can do something like that:
>
> a={s:'string', l: 0l}
> print, n_tags(a)
> print, tag_names(a)
> help, a, /str
> etc.
>
> But if I've an object I can't do the same... Is it possible to parse
> all the attributes inside the object using something similar ?
> I mean, I don't want to use the GetProperty method, I only want to
> know which attributes are stored in the definition of the object.
>
> pro object__define
> str ={ object, $
> attribute_s: '', $
> attribute_l: 0l }
> end
>
> The object saves a struct. It could be possible to take information
> about this without the use of the getproperty method.

```
a = Obj_New('object')
HELP, Create_Struct(NAME=Obj_Class(a)), /Structure
```

Cheers,

David

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

Subject: Re: Treating an object as a structure
Posted by natha on Wed, 04 Mar 2009 21:26:18 GMT
View Forum Message <> Reply to Message

OK,

It works but the problem is that the struct is not the object itself.
Doing aa=create_struct(name=obj_class(obj)) creates a struct with the
same fields as object attributes but all of these attributes are not
uninitialized.

Subject: Re: Treating an object as a structure
Posted by David Fanning on Wed, 04 Mar 2009 21:59:42 GMT
View Forum Message <> Reply to Message

llo writes:

> It works but the problem is that the struct is not the object itself.

No, that's pretty much the point of objects. :-)

> Doing aa=create_struct(name=obj_class(obj)) creates a struct with the
> same fields as object attributes but all of these attributes are not
> uninitialized.

The point of objects is to *encapsulate* the data.
In other words, keep it hidden from prying eyes.
Perhaps it is not an object you want. Can you do
what you want to do in front of God and all people
with a structure? If so, I'd say use that.

I realize you don't want to write a method, but how
about something simple like this:

```
pro object::help
   struct = Create_Struct(NAME=obj_class(self))
   tags = Tag_Names(struct)
   for j=0,n_elements(tags)-1 do begin
      Help, self.(j), Output=out
      print, tags[j] + ' :  ' + StrMid(out, 13)
    endfor
end

function object::init
  self.attribute_s = 'This is an atribute'
  self.attribute_l = 456
  return, 1
end

pro object__define
       str ={ object, $
           attribute_s: '', $
           attribute_l: 0l  }
```

```
end

  a = Obj_New('object')
  a -> Help
END
```

When I run this I get:

```
ATTRIBUTE_S :    STRING    = 'This is an atribute'
ATTRIBUTE_L :    LONG      =          456
```

Cheers,

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

Subject: Re: Treating an object as a structure
Posted by natha on Wed, 04 Mar 2009 23:05:02 GMT
View Forum Message <> Reply to Message

Ok, thanks ...

With this discussion, all of a sudden I wondered how ITT people implement the ALL keyword in GetProperty method of the objects ? Because maybe the easiest way to get all the content of the object is having the ALL keyword in GetProperty method. (I know I said I don't wanna use this method)

I don't know how can I implement this method. Maybe I will try something like this:

```
PRO object::GetProperty, A=a, B=b, C=c, D=d, E=e, F=f, G=g, ALL=all

....

  IF ARG_PRESENT(all) THEN BEGIN
    all=CREATE_STRUCT
 ('A',self.a,'B',self.b,'C',self.c,'D',self.d,'E',self.e,'F', self.f,'G',self.g)
  ENDIF
END
```

Is it correct or there are a better way to do that ?

Subject: Re: Treating an object as a structure
Posted by David Fanning on Wed, 04 Mar 2009 23:39:29 GMT
View Forum Message <> Reply to Message

llo writes:

> Is it correct or there are a better way to do that ?

I don't know. An ALL keyword always seemed like too much
work to me. I've never done it. :-(

Cheers,

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

Subject: Re: Treating an object as a structure
Posted by Michael Galloy on Wed, 04 Mar 2009 23:45:16 GMT
View Forum Message <> Reply to Message

llo wrote:
> Ok, thanks ...
>
> With this discussion, all of a sudden I wondered how ITT people
> implement the ALL keyword in GetProperty method of the objects ?
> Because maybe the easiest way to get all the content of the object is
> having the ALL keyword in GetProperty method. (I know I said I don't
> wanna use this method)
>
> I don't know how can I implement this method. Maybe I will try
> something like this:
>
> PRO object::GetProperty, A=a, B=b, C=c, D=d, E=e, F=f, G=g, ALL=all
>
> ....
>
> IF ARG_PRESENT(all) THEN BEGIN
> all=CREATE_STRUCT
> ('A',self.a,'B',self.b,'C',self.c,'D',self.d,'E',self.e,'F', self.f,'G',self.g)
> ENDIF
> END
>
> Is it correct or there are a better way to do that ?

Just use STRUCT_ASSIGN:

```
pro mg_instance_test::getProperty, all=all
  compile_opt strictarr

  if (arg_present(all)) then begin
    all = create_struct(name=obj_class(self))
    struct_assign, self, all
  endif
end


function mg_instance_test::init
  compile_opt strictarr

  self.a = 3.0
  self.b = 'Hello'
  self.c = 123L

  return, 1
end


pro mg_instance_test__define
  compile_opt strictarr

  define = { mg_instance_test, a: 0.0, b: '', c: 0L }
end
```

Run the example like:

```
  IDL> o = obj_new('mg_instance_test')
  IDL> o->getProperty, all=all
  IDL> help, all, /structures
  IDL> obj_destroy, o
```

Mike
--
www.michaelgalloy.com
Associate Research Scientist
Tech-X Corporation

Subject: Re: Treating an object as a structure
Posted by natha on Thu, 05 Mar 2009 15:36:55 GMT

On Mar 4, 4:59 pm, David Fanning <n...@dfanning.com> wrote:
> llo writes:
>> It works but the problem is that the struct is not the object itself.
>
> No, that's pretty much the point of objects. :-)

That's not true David. You can get all the object attributes doing
this:

```
a=OBJ_NEW('object')
str=CREATE_STRUCT(NAME=OBJ_CLASS(a))
STRUCT_ASSIGN, a, str
```

Now, ypu have all the attributes of a in the struct str.

Cheers,

Bernat