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.")