Subject: Re: how to get info on object fields
Posted by davidf on Mon, 25 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Mirko Vukovic writes:

> I was trying to write a general extraction routine for objects. The
> idea was to pass the name of a field, it would check if the field
> exists, get its tag number and extract it. It would work on the object
> as if it were a structure. But TAG_NAMES does not work on objects
> (which does make sense as SELF is still an object (type =11), not a
> structure). (see code below for my implementation).
>
> I like using objects, but sometimes they are too cumbersome for
> debugging purposes. I would also like to have a set of general routines
> applicable to all objects (via inheritance). That is why a general
> routine for extracting a field would be nice. Any clues how I might go
> about it?

I used this code to get something working in just a few minutes.
These objects can have superclasses, of course, and I am not
checking for that. You could use Obj_Class to find all the
superclass structures in an iterative way. The code would not
be too hard to implement.

The way this code works is to use Obj_Class to get the
name of the object, which is the name of the structure
you want. I create the right kind of structure with
the Execute command. To extract the field, I have to
use a second Execute statement.

```
FUNCTION OBJECT::EXTRACT, field

; Check if FIELD is a valid field name. If it is,
; return the field. If not, return -1.

thisClass = Obj_Class(self)
ok = Execute('thisStruct = {' + thisClass + '}')
structFields = Tag_Names(thisStruct)
index = WHERE(structFields EQ StrUpCase(field), count)
IF count EQ 1 THEN BEGIN
   ok = Execute('Return, SELF.' + structFields(index(0)))
ENDIF ELSE BEGIN
   Message, 'Can not find field "' + field + $
      '" in structure.', /Informational
   RETURN, -1
ENDELSE
END
```

This should give you some ideas.

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
Customizable IDL Programming Courses
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com