

---

Subject: how to get info on object fields

Posted by [Mirko Vukovic](#) on Mon, 25 Aug 1997 07:00:00 GMT

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

---

Hi,

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?

```
FUNCTION structured::extract,arg,depointer = depointer

; check if arg is a valid field name
  ii = where(tag_names(self) EQ strupcase(arg),count)
  IF count NE 0 THEN begin
    type = size(self.(ii))
    type = type[type[0]+1]
    IF keyword_set(depointer) THEN if type EQ 10 THEN
return,*self.(ii)
    return,self.(ii)
  ENDIF ELSE BEGIN
    message,/info,'Invalid field name provided'
    print,'You specified'+arg
    print,'Available fields are:'
    print,tag_names(self)
    return,0
  ENDELSE
```

--

Mirko Vukovic, Ph.D 3075 Hansen Way M/S K-109  
Novellus Systems Palo Alto, CA, 94304  
415/424-4969 mirko.vukovic@varian.grc.com

---