Subject: Smart way to extract the same attribute from a list of objects?
Posted by Paul Van Delst[1] on Thu, 04 Oct 2012 15:44:07 GMT

View Forum Message <> Reply to Message

Hello,

I have a list,

```
IDL> help, r
R               LIST  <ID=2  NELEMENTS=616>
```

composed of only a single type of object,

```
IDL> help, r[0]
<Expression>    OBJREF    = <ObjHeapVar3(RTSOLUTION)>
IDL> help, r[200]
<Expression>    OBJREF    = <ObjHeapVar1803(RTSOLUTION)>
..etc..
```

What I want to do is extract the same attribute from every object in the list.

The way I'm doing it now is:

```
  radiance = DBLARR(r.Count())
  FOR i=0,r.count()-1 DO BEGIN
    r[i].get_property, radiance=x
    radiance[i]=x
  ENDFOR
```

and then I can do stuff with "radiance", such as plot it.

I find the above a little bit clunky since I have to explicitly iterate through the list to pull out what I want, creating the temporary "holding" array beforehand. If I want to extract more than one thing I end up with something like

```
  n = r.Count()
  channel  = LONARR(n)
  radiance = DBLARR(n)
  FOR i=0,n-1 DO BEGIN
    r[i].get_property, radiance=x, channel=j
    radiance[i]= x
    channel[i] = j
  ENDFOR
```

which just looks messy.

Does anyone have any better/faster/more-elegant solutions to this sort
of problem?

I've had a look at the _overloadBracketsRightSide function documentation
but, to be honest, the description seems rather impenetrable to me. I
guess I'm looking for something akin to the "ELEMENTAL" attribute in
Fortran where a procedure written for in/output scalars can also operate
on any rank arrays (as long as they are conformable.)

cheers,

paulv

Subject: Re: Smart way to extract the same attribute from a list of objects?
Posted by PMan on Thu, 04 Oct 2012 16:56:44 GMT
View Forum Message <> Reply to Message

Actually, I my setProperty and getProperty functions I stole from here:

http://idldatapoint.com/2012/01/05/a-technique-for-flexible- getsetproperty-methods-in-idl-8/

The author did something similar to me, but uses a hash object inside his object, whereas I just inherited the list object into my object.