Subject: Re: How to get _overloadSize to return N_Dimensions=0?
Posted by Michael Galloy on Wed, 19 Apr 2017 21:19:40 GMT
View Forum Message <> Reply to Message

On 4/18/17 7:56 AM, Matthew Argall wrote:
> The problem with undefined variables was fixed, but there is another
> bug related to scalar values. I assume N_Elements is determined as
> Product(Size(input, /DIMENSIONS)). However, a scalar value has a
> dimension size of 0, so the result of 0 elements.
>
>
> function test_olSize::_OverloadSize
> return, size(*self.value, /DIMENSIONS)
> end
>
> function test_olSize::Init, value
> compile_opt strictarr
> self.value = Ptr_New(value, /NO_COPY)
> return, 1
> end
>
> pro test_olSize__define
> class = {test_olSize, $
> inherits IDL_Object, $
> value: ptr_new()}
> end
>
>
> IDL> myObj = Obj_New('Test_olSize', 1)
> IDL> Print, N_Elements(myObj)
> 0
>

I would use something like the following for your _overloadSize method:

```
function test_olsize::_overloadSize
  return, size(*self.value, /n_dimensions) eq 0 $
          ? 1L $
          : size(*self.value, /dimensions)
end

IDL> myObj = Obj_New('Test_olSize', 1)
IDL> Print, N_Elements(myObj)
          1
```

Mike

--