
Subject: How to get `_overloadSize` to return `N_Dimensions=0`?

Posted by [Matthew Argall](#) on Tue, 27 May 2014 22:40:41 GMT

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

I was wondering how it is possible to get the `_overloadSize` function to return `N_DIMENSION=0` when there are no dimensions. Below is an example. The docs seems like it reports `N_Elements(Size(input, /DIMENSIONS))`, which is not correct for an undefined variable...

```
IDL> print, size(!Null, /N_DIMENSIONS)
```

```
0
```

```
IDL> .run test_olSize__define
```

```
Dimensions:  [0]
```

```
N_Elements:  0
```

```
N_Dimensions: 1
```

```
;-----  
function test_olSize::_OverloadSize  
    return, size(*self.value, /N_DIMENSIONS)  
end
```

```
function test_olSize::Init  
    compile_opt strictarr  
    self.value = Ptr_New(/ALLOCATE_HEAP)  
    return, 1  
end
```

```
pro test_olSize__define  
    class = {test_olSize, $  
        inherits IDL_Object, $  
        value: ptr_new()  
    }  
end
```

```
;Main level test program  
myObj = Obj_New('Test_olSize')  
print, 'Dimensions:  ', '[' + strjoin(strtrim(size(myObj, /DIMENSIONS), 2), ', ' ) + ']'  
print, 'N_Elements:  ', strtrim(size(myObj, /N_ELEMENTS), 2)  
print, 'N_Dimensions: ', strtrim(size(myObj, /N_DIMENSIONS), 2)  
obj_destroy, myObj  
  
end
```
