Subject: Re: Weird MIN behavior
Posted by Michael Wallace on Mon, 21 Feb 2005 01:12:21 GMT

View Forum Message <> Reply to Message

Mark Hadfield wrote:

```
> Michael Wallace wrote:
>
>> Can someone explain this?
>>
>>
>> Normal case; everything works as you expect.
>>
>> IDL> a = [1.0, 2.0, 3.0, 4.0, 5.0]
>> IDL> b = min(a, MAX = c)
>> IDL> print, b, c
>>        1.00000       5.00000
>>
>>
>> Abnormal case; why doesn't this work?
>>
>> IDL> a = [1.0, 2.0, 3.0, 4.0, 5.0]
>> IDL> r = dblarr(2)
>> IDL> r[0] = min(a, MAX = r[1])
>> % MIN: Expression must be named variable in this context: <DOUBLE   ((
>>        0.00000,      0.00000))>.
>
>
> It's complaining about the argument passed to the MAX keyword. As it
> says, this must be a named variable, and a subscripted variable does not
> qualify because it is passed by value. You're lucky that IDL tells you
> that it doesn't accept a subscripted variable here--in other situations
> it quitely leaves the value unmodified and let's you puzzle it out later.
```

Thanks. I had no clue what the message meant by "named variable." You wouldn't believe how long I stared at that error message trying to figure out what it meant. I knew things would work when I used non-subscripted variables, but the message still didn't make sense. Now, the message could have said "Look you idiot, a subscripted variable is sent by value, so the function can't modify the value. This probably isn't what you want." Now that makes sense. ;-)

I guess I had gotten it into my head that since you can change primitive values within a function by using KEYWORD = value, you must also be able to change array elements as well. It shouldn't surprise me that IDL is backwards. In every other language, arrays and array elements are passed by reference and primitives are passed by value.

This will get added to my ever growing list of gotchas.

-Mike