Subject: Re: Defining Integer Field in a Structure
Posted by Craig Markwardt on Thu, 01 Nov 2001 23:36:38 GMT
View Forum Message <> Reply to Message

JD Smith <jdsmith@astro.cornell.edu> writes:

> Brian Bell wrote:
>>
>> Is there a way to define a field within a structure whose type is
>> integer? In my case, I want the field to be any integer from 0 to 4.
>> I know you can assign an actual integer to the field, but I don't want
>> to do this because I want the field value to change within certain
>> procedures and to return that value to the procedures calling it. I
>> would appreciate any help you might be able to offer.
>>
>
> This is illegal:
>
> IDL> a={0:"this", 1:"is",2:"a",3:"funny",4:"test"}
>
> so the short answer is no. However, if you just want 4 numbered
> "slots", try an array:
>
> IDL> a={vals:['this','is','a','funny','test']}
> IDL> print,a.vals[1]
> is
>
> if you'd like the types of each slot to be different, try at a pointer
> array. If you'd like the number of "slots" to be variable, try a
> pointer to a pointer array.

Following up on this diversion, it was in fact possible in IDL 5.2 and earlier to make structures whose tag names were not valid IDL variable names. For example, this statement was perfectly valid and executable:

```
x = create_struct('1',"this",'2',"is",'3',"funny",'4',"test ")
```

and it made exactly the structure you were expecting. However, I'm not sure it would be useful, and more importantly these types of antics are now forbidden by the IDL runtime.

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

---------- ----------