Subject: Re: Error in concatenating structure arrays
Posted by davidf on Wed, 30 Apr 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Paul van Delst <paulv@airs2.ssec.wisc.edu> writes:

> The following code :

> pro test
> as_state = { x0:-1L, y0:-1L, x1:-1L, y1:-1L, select:-1L, color:-1L }
> info = { as_state : as_state }
> dummy = { x0:2L, y0:2L, x1:2L, y1:2L, select:2L, color:2L }
> info = { as_state : [ info.as_state, dummy ] }
> help, info.as_state
> end
>
> gives the result :
>
> % Conflicting data structures: DUMMY,MISSING.
>
> i.e. when I try to concatenate the structures, it fails.
> The following code defines the dummy structure differently:
>
> pro test
> as_state = { x0:-1L, y0:-1L, x1:-1L, y1:-1L, select:-1L, color:-1L }
> info = { as_state : as_state }
>
> dummy = info.as_state
> dummy.x0 = 2L
> dummy.y0 = 2L
> dummy.x1 = 2L
> dummy.y1 = 2L
> dummy.select = 2L
> dummy.color = 2L
> info = { as_state : [ info.as_state, dummy ] }
> help, info.as_state
> end
>
> and works as expected with the result:
>
> <Expression> STRUCT = -> <Anonymous> Array(2)
>
> My question is: Why does the first method, where the dummy structure to
> be concatenated is explicitly defined, not work? What does the error
> message actually mean?

I think what this message means is that two anonymous
structures, even though they are defined similarly, are

not the same. That is, an anonymous structure when it is defined is given a "defined" identity internally. In other words, IDL has some way of "knowing" about it. To prove that this must be the case, consider this:

```
a = { array:FltArr(10) }
b = FltArr(20)
a.array = b
```

This code causes the same (unhelpful) error message as above:

```
% Conflicting data structures: B,MISSING.
```

The reason being, I think, that IDL "knows" about the structure A, knows that the field array is 10 elements in length, and can't allow something bigger to fit into the space that has been allocated in memory. In some sense, the structure A is defined internally as a specific entity. (This would have to be the case if you think about it.)

Arrays, of course, must be composed of *identical* elements. In your first example, info.as_state and dummy are two different entities (even though they are defined alike) and cannot be put into the same array.

In the second case, dummy is an *instance* of info.as_state and, hence, is *identical* to it from an internal structure definition or entity point of view. Thus, you can put dummy into an array with the info.as_state structure.

Cheers!

David

---

David Fanning, Ph.D.
Fanning Software Consulting
Customizable IDL Programming Courses
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com