Subject: Re: Q: Array of Structures
Posted by Vap User on Wed, 17 Jun 1998 07:00:00 GMT
View Forum Message <> Reply to Message

korpela@islay.ssl.berkeley.edu (Eric J. Korpela) writes:

One may also use anonymous structures, as Dave did in after 'but now, I try this'. I concatenate anonymous structures frequently, so when I saw Dave' assertion that you can't do it, I shook my head in wonder. However, I found out, only too quickly, that all of you are right. So, how do I do it? The secret is that at the beginning of whatever program I'm doing the concatenation I define the anonymous structure and a blank copy of it. I keep filling the blank copy with data and then concatenating it to create the larger array of structures. Like this.

Start program

```
  a={some anonymous structure definition} & b=a
  start your loop construct here
 do your processing
 fill in b structure
       if first iteration
         output_array_of_structures=b
 else
   output_array_of_structures = [output_array_of_structures,b]
  end loop
end program
```

No conflicts that I can see.

You could use named structures, with the additional hassle, perhaps small in your case, of having to exit IDL everytime you make a change to the structure.

I suspect it has to do with an underlying 'naming' convention for anonymous structures, (do a help,/structure. See the '** Structure <long hex number>' The long hex number is the 'type' I believe) so that deep down in IDLs code there is some comparison of the structure 'types' going on, not comparison of the actual contents of the structure.

I would point out that if you know how big your output will be, you should just go ahead and create the array in the beginning of the processing. Concatenation is a *slow* process. Alternately, you can create the largest array you think you'll need then enlarge it if your processing produces more output than expected.

>
> In article <MPG.feb2857db08f77a989681@news.frii.com>,
> David Fanning <davidf@dfanning.com> wrote:
>> But, look. I just tried this:
>>
>>   a = { data:FltArr(10), name:'' }
>>   b = { data:FltArr(10), name:'' }
>>   c = [a, b]
>>   % Conflicting data structures: B,concatenation.
>>   % Execution halted at: $MAIN$
>>
>> But now, I try this:
>>
>>   a = { data:FltArr(10), name:'' }
>>   b = a
>>   c = [a, b]
>>
>> No problem!
>
> I think that the problems it that the first example creates anonymous
> structures for each variable. Because the structures won't have the
> same tag (anonymous structures are still tagged invisibly), they are
> treated as different structure types. The problem goes away for tagged
> structures. The following works just fine.
>
> a = {StruTag, data:FltArr(10), name:'' }
> b = {StruTag, data:FltArr(10), name:'' }
> c = [a, b]
>
> In the second example, the assignment gives b the same structure tag
> as a, so the concatenation works just fine.
>
> Personally, I'd like to see IDL check for identical anonymous structures
> and set the structure tags appropriately.
>
> Eric
> --
> Eric Korpela | An object at rest can never be
> korpela@ssl.berkeley.edu | stopped.
> Click for home page.

--
I don't speak for JPL, it doesn't speak for me.
Well, not all the time, at least.
William Daffer <vapuser@haifung.jpl.nasa.gov>