Subject: Re: newbie question / concatenation of arrays of nested structure
Posted by Thomas A. McGlynn on Wed, 08 Sep 1999 07:00:00 GMT
View Forum Message <> Reply to Message

If you don't want to modify the code that is creating the structures
to be concatenated then neither of these approaches might be feasible. You
could then do something like:

```
first = replicate({...}, ...)
....
second = replicate({...}, ...)
....

temp = replicate(first[0], n_elements(second));
for i=0, n_tags(first[0])-1 do temp.(i) = second.(i)
first = [first,temp]
```

This still isn't too bad, just three lines for a reasonably generic solution.

So long as the types agree you don't even have to worry about the
structure tag names matching. One could make build this into a function
easily enough -- though a general routine would probably need to deal
with recursive structures intelligently (and maybe multi-dimensional
arrays of structures).

Regards,
Tom McGlynn

Liam Gumley wrote:
>
> Liam Gumley wrote:
>> The only way to create equivalent structures is to use named structures,
>> e.g.
>>
>> IDL> record = {z, a:0, b:'name', c:0}
>> IDL> first = replicate(record, 5)
>> IDL> second = replicate(record, 3)
>> IDL> combo = [first, second]
>
> Note to self: Any time you say "The only way" in this newsgroup, you're
> bound to be wrong.
>
> David's web page correctly points out that copies of an anonymous
> structure are equivalent, and thus can be concatenated, e.g.
>
> IDL> record = {a:0, b:'name', c:{d:0, e:0}}
> IDL> a = record
> IDL> b = record

> IDL> c = [a, b]
>
> Cheers,
> Liam.
>
> --
> Liam E. Gumley
> Space Science and Engineering Center, UW-Madison
> http://cimss.ssec.wisc.edu/~gumley
