Subject: Re: resizing an array of structures (uugh)
Posted by davidf on Mon, 04 Jun 2001 13:43:11 GMT

Randall Skelton (rhskelto@atm.ox.ac.uk) writes:

> I have an ascii file containing a few thousand lines with each individual
> dataset comprising 10 lines of floats, ints, strings, etc. It seems
> logical to read this in as an array of structures as each dataset contains
> the same information, just different numbers.
>
> My problem is that I don't know what the dimension of the array should be
> before I start. Initially I just defined a large array and counted the
> number of datasets for subsequent processing. However, as time progresses
> and this code gets more use, I have to say that I really hate all the
> excess array elements... I figured there would be an easy way to resize
> the array of structures, but the best I can come up with is a double for
> loop that is rather slow.
>
> ; loop over the number of array elements
> for i, n_elements(array) do begin
>   ; loop over the number of tags
>   for j, n_tags(structure) do begin
>   resized_array[i].(j) = array[i].(j)
>   endfor
> endfor
>
> Is there a *faster* or more elegant way to do this? Does IDL have a
> *fast* resize command that can handle any type of array to simply adjust
> the number of elements in the array without rebinning, or otherwise
> changing the numbers?

I've read your article several times, and I still can't
convince myself I'm not missing something. (Is that
a double negative!?) But I don't see any reason you
can't treat an array of structures in the same way
you treat an array of anything. Since you are counting
the structures as you fill them into the array, don't
you just want this:

   array = array[0:count-1]

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting

Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155