Subject: Re: Splitting An Array Of Strings Without Using Loops
Posted by Paul Van Delst[1] on Mon, 28 Jul 2003 18:47:21 GMT
View Forum Message <> Reply to Message

Darrick White wrote:

```
>
>> Dear Darrick,
>>
>> here is a second solution using reads.
>>
>> pro test
>> data=['1:23','2:32','3:30','4:45']
>>
>> s={x:bytarr(1),s:bytarr(1),y:bytarr(2)}
>> s=replicate(s,4)
>>
>> reads,byte(data),s
>>
>> print,string(s.x)
>> print,string(s.y)
>> end
>>
>> IDL> 1 2 3 4
>> IDL> 23 32 30 45
>
> It looks like I'm not explaining my problem clearly.  For instance,
> the following sets of data are valid inputs to my application:
>
> 1) data=['1:23','2:32','3:30','4:45']
> 2) data=['12:23','22:32:34:45','32:30','42:45:90']
> 3) data=['100:23','200:32','300:30','400:45']
> 4) data=['1:23:2','2:32:2','3:30:2','4:45:2']
>
> The resulting transformation would like this for both:
>
> 1) print, intarr(2,4)
>    1 23
>    2 32
>    3 30
>    4 45
>
> 2) print, intarr(4,4)
>    12 23 NaN NaN
>    22 32 34 45
>    32 30 NaN NaN
>    42 45 90 NaN
```

Wouldn't this need to be a two-pass problem? You parse the input data to determine the individual entry and maximum dimension (in this case 4 due to the 22:32:34:45), create you array with fill values, and then "go through the array once more" to fill in your array. (The quotes are there because going through the array once more could be achieved a number of ways.)

I would think that smart usage of the IDL string functions should be able to do most of that sans looping. (Otherwise, I'm sure JD can come up with some neato supa-quick method using HISTOGRAM.... :o)

paulv

p.s. If you're only using integers, you can't use NaN as a fill value.

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7748
Fax:(301)763-8545