Subject: Splitting An Array Of Strings Without Using Loops
Posted by darrick.white on Fri, 25 Jul 2003 04:42:56 GMT
View Forum Message <> Reply to Message

This is probably simple, but I'm having a time trying to figure it out. I want to be able to split an array of strings without using loops.

Example:
dataPoints is an array of strings with N elements
The format of each element within dataPoints is "x:y1:y2:y3:yn". More than likely, the data will be in the format of x:y".

This array will become data points (the first element is always considered the x coordinate): (x,y) = 1,23. In case of multiple points (2:21:34:54), the data will look like: (2,21), (2,34), (2,54).

I need a way to take:
dataPoints[0] = 1:23
dataPoints[1] = 2:32
dataPoints[2] = 3:30
dataPoints[3] = 4:45

and create
points[2,4]
1 23
2 32
3 30
4 45

-Darrick

Subject: Re: Splitting An Array Of Strings Without Using Loops
Posted by R.Bauer on Sat, 26 Jul 2003 11:10:04 GMT
View Forum Message <> Reply to Message

Darrick White wrote:

> This is probably simple, but I'm having a time trying to figure it
> out. I want to be able to split an array of strings without using
> loops.
>
> Example:
> dataPoints is an array of strings with N elements
> The format of each element within dataPoints is "x:y1:y2:y3:yn". More
> than likely, the data will be in the format of x:y".

>
> This array will become data points (the first element is always
> considered the x coordinate): (x,y) = 1,23. In case of multiple
> points (2:21:34:54), the data will look like: (2,21), (2,34), (2,54).
>
> I need a way to take:
> dataPoints[0] = 1:23
> dataPoints[1] = 2:32
> dataPoints[2] = 3:30
> dataPoints[3] = 4:45
>
>
> and create
> points[2,4]
> 1 23
> 2 32
> 3 30
> 4 45
>
> -Darrick

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
```

--
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg-i/

Subject: Re: Splitting An Array Of Strings Without Using Loops
Posted by darrick.white on Mon, 28 Jul 2003 17:17:27 GMT
View Forum Message <> Reply to Message

> 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

3) print, intarr(2,4)
  100 23
  200 32
  300 30
  400 45

4) print, intarr(3,4)
  1 23 2
  2 32 2
  3 30 2
  4 45 2

Is there a way (not knowing what data set input is used) to transform my data into the corresponding result array? Note: For transformation #2 above, I need to append each point to my new array. If the array dimensions don't match, I need to fill in those missing elements with 'NaN'.

Thanks
-Darrick

Subject: Re: Splitting An Array Of Strings Without Using Loops
Posted by David Fanning on Mon, 28 Jul 2003 17:23:40 GMT
View Forum Message <> Reply to Message

Darrick White writes:

> Is there a way (not knowing what data set input is used) to transform
> my data into the corresponding result array?

Without knowing anything about the input data!?
I'd try prayer, but that's just me. :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Splitting An Array Of Strings Without Using Loops
Posted by Rick Towler on Mon, 28 Jul 2003 18:24:50 GMT
View Forum Message <> Reply to Message

"Darrick White" wrote...

> It looks like I'm not explaining my problem clearly.

> Is there a way (not knowing what data set input is used) to transform
> my data into the corresponding result array?

I don't think the issue is one of clarity, but of possibility. Unless JD
can save you with some magical incarnation of HISTOGRAM you are going to
have to change your design criteria or use a loop. If performance is really
that important write this function in C.

-Rick

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

Subject: Re: Splitting An Array Of Strings Without Using Loops
Posted by JD Smith on Mon, 28 Jul 2003 23:00:33 GMT

View Forum Message <> Reply to Message

On Mon, 28 Jul 2003 11:24:50 -0700, Rick Towler wrote:

> "Darrick White" wrote...
>

>> It looks like I'm not explaining my problem clearly.
>
>> Is there a way (not knowing what data set input is used) to transform
>> my data into the corresponding result array?
>
> I don't think the issue is one of clarity, but of possibility. Unless
> JD can save you with some magical incarnation of HISTOGRAM you are going
> to have to change your design criteria or use a loop. If performance is
> really that important write this function in C.
>
> -Rick

Come on people. I don't use HISTOGRAM for everything. I use it very rarely, in fact.

How about something like:

```
nums=strsplit(strjoin(data,':'),':',/EXTRACT)
cnts=long(total(byte(data) eq 58b,1))+1L
```

Now you have a list of tuple-counts and the tuples themselves in a long list. You could (yes) use HISTOGRAM or perhaps many other methods to stick these into an array as you describe without looping, but rather than show something you'd forget 5 minutes after dropping it into your code, I'll join Rick in saying that if parsing these strings quickly is this important to you, you'll get better results by re-designing the input format, or pre-parsing them using a language better suited to these manipulations. And on the off chance that you're suffering from the "must-optimize-everything-in-sight" disease, you'll want to make sure a readable and straightforward input loop won't meet your needs before venturing too far into IDL esoterica:

```
b=make_array(/LONG,VALUE=-1,max(cnt),n_elements(data))
for i=0,n_elements(data)-1 do b[0,i]=strsplit(data[i],':',/EXTRACT)
```

Note that there's no integer (long or otherwise) definition of NaN, so I used -1.

JD

Subject: Re: Splitting An Array Of Strings Without Using Loops
Posted by Pavel Romashkin on Tue, 29 Jul 2003 15:46:00 GMT
View Forum Message <> Reply to Message

JD Smith wrote:
>

> Come on people. I don't use HISTOGRAM for everything. I use it very
> rarely, in fact.

We don't belive this! Now that you've got the reputation, there is no
getting away from it :-)

> You could (yes) use HISTOGRAM or perhaps many other methods

See? Told you! :-)

Cheers,
Pavel

Subject: Re: Splitting An Array Of Strings Without Using Loops
Posted by JD Smith on Tue, 29 Jul 2003 16:24:25 GMT
View Forum Message <> Reply to Message

On Tue, 29 Jul 2003 08:46:00 -0700, Pavel Romashkin wrote:

> JD Smith wrote:
>>
>> Come on people. I don't use HISTOGRAM for everything. I use it very
>> rarely, in fact.
>
> We don't belive this! Now that you've got the reputation, there is no
> getting away from it :-)
>
>> You could (yes) use HISTOGRAM or perhaps many other methods
>
> See? Told you! :-)

You could use HISTOGRAM. I've given it up for the month.

JD

Subject: Re: Splitting An Array Of Strings Without Using Loops
Posted by Pavel Romashkin on Tue, 29 Jul 2003 18:10:25 GMT
View Forum Message <> Reply to Message

Okay, that means you are getting back to it in two days :-)

Pavel

JD Smith wrote:
>

> *You* could use HISTOGRAM. I've given it up for the month.
>
> JD