
Subject: Re: Splitting An Array Of Strings Without Using Loops
Posted by [R.Bauer](#) on Wed, 30 Jul 2003 18:25:45 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']
>
```

ok

do the followings. reads is very powerful (not only histogram)

```
data=replace_string(data,':','')
```

```
1) data=replace_string(data,':','')
```

```
    a=intarr(2)
    b=intarr(2)
    c=intarr(2)
    d=intarr(2)
reads,data,a,b,c,d
print,a,b,c,d
```

```

1  23
2  32
3  30
4  45

```

2) data=replace_string(data,':','')

```

a=intarr(2)
b=intarr(4)
c=intarr(2)
d=intarr(3)
reads,data,a,b,c,d
print,a,b,c,d
12  23
22  32  34  45
32  30
42  45  90

```

3) data=replace_string(data,':','')

```

a=intarr(2)
b=intarr(2)
c=intarr(2)
d=intarr(2)
reads,data,a,b,c,d
print,a,b,c,d
100  23
200  32
300  30
400  45

```

4) data=replace_string(data,':','')

```

a=intarr(3)
b=intarr(3)
c=intarr(3)
d=intarr(3)
reads,data,a,b,c,d
print,a,b,c,d
1  23  2
2  32  2
3  30  2
4  45  2

```

You'll find replace_string at our library:

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl_html/dbase/download/replace_string.tgz

or as idl5.6 binary

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl_html/dbase/download/replace_string.sav

Please have a look for licensing and further routines at

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro.html

If you write yourself a function to count the ":" signs +1 then this could be easy automated.

regards

Reimar

> 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

--

Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro.html

=====