

---

Subject: Re: 'Replicat' a string array  
Posted by [Heinz Stege](#) on Thu, 19 Dec 2013 15:40:46 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Thu, 19 Dec 2013 06:25:20 -0700, David Fanning wrote:

```
> Heinz Stege writes:
>
>>
>> On Wed, 18 Dec 2013 07:02:27 -0700, David Fanning wrote:
>>
>>> Maybe something like this:
>>>
>>> IDL> a = '2008-07-25T20:56' & b = '2008-07-25T21:32'
>>> IDL> c = [Reform(Replicate(a, 20), 1, 20), $
>>>      Reform(Replicate(b,20), 1, 20)]
>>> IDL> help, c
>>> C          STRING   = Array[2, 20]
>>>
>> or this:
>> IDL> a=['2008-07-25T20:56', '2008-07-25T21:32']
>> IDL> c=a[* ,intarr(100)]
>> IDL> help,c
>> C          STRING   = Array[2, 100]
>>
> OK, I just got back from cutting trees. How in God's name does *this*
> work,
```

I think the explanation can be found in the combination of two features from the IDL language. The first is documented in the chapter "Using Arrays as Array Subscripts" of the IDL Manual. You can write

```
y=x[[i,j]]
```

to get the i-th and the j-th element of the array x. No one can stop you to write

```
y=x[[i,i]]
```

to get the i-th element doubly. And, because [i,i] is nothing else than an array, you can write

```
y=x[intarr(n)+i]
```

to get the i-th element n times. For i=0 it is

```
y=x[intarr(n)]
```

The second feature is the use of "Extra dimensions". (See "Understanding Array Subscripts" in the manual.) In my words every IDL array has 8 dimensions. All trailing dimensions of size 1 are automatically removed by IDL, but can be referenced by "0" or "\*". You may write

```
c=a[* ,0] or c=a[* ,0,0,0,0,0,0,0]
```

to reproduce the string array of the OP.

Combining both features results in

```
c=a[:,intarr(n)]
```

That's it.

By the way, it saves some memory using

```
c=a[:,bytarr(n)]
```

instead of the expression above. I was careless when quick replying to the posting.

Another note: The above approach is reasonable for string arrays only.

For integer and floating point arrays it is more useful to use the

rebin function. See message ID b073e6\$5p5\$1@newsreader.wustl.edu (or

<https://groups.google.com/forum/?hl=en#!topic/comp.lang.idl-pvwave/VWBB8kSs0HU>

for google users) for an earlier discussion within this group. The

rebin function saves memory and CPU time for very large integer or

float arrays. However, rebin does not work for string arrays.

> and how did you ever stumble onto it?

Good question. I think, when I started working with PV-Wave more than

20 years ago, I didn't realize the benefit of the rebin function and

found the solution above...

> Strangest thing I have ever

> seen in IDL, and I've seen a LOT of strange things!

Thank you.

Cheers, Heinz

---