

---

Subject: Re: string manipulation

Posted by [John-David T. Smith](#) on Tue, 27 Feb 2001 22:40:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Wayne Landsman wrote:

>  
> In article <on66hxm5tk.fsf@cow.physics.wisc.edu>, craigmnet@cow.physics.wisc.edu writes...  
>> This is primarily because STRMID and STRPUT are not  
>> vectorized at all. Well STRMID \*is\* vectorized, but not with a sane  
>> behavior. For example, what I'd like to do is:  
>>  
>> NEWKEY = STRMID(KEY,0,P1) + '50' + STRMID(KEY,P2,100)  
>> NEWKEY = STRMID(KEY,0,P1) + '50' + STRMID(KEY,P2,100)  
>>  
>> Where KEY, P1, and P2 are vectors. Obviously this doesn't work. Any  
>> ideas?  
>  
> The problem when STRMID was vectorized for V5.3 was that it was made \*too\*  
> powerful -- it handles simultaneously both extraction from multiple strings  
> and multiple extractions from a single string. In practice, I think the  
> first situation -- extraction from multiple strings -- is far more common, but  
> has an ugly syntax in the current STRMID implementation. Here is how one  
> does the example above.  
>  
> N = N\_elements(KEY)  
> NEWKEY = STRMID(KEY,INTARR(1,N),REFORM(P1,1,N)) + '50' + \$  
> STRMID(KEY,REFORM(P2,1,N), REPLICATE(100,1,N) )  
>  
> I have thought about writing a simple wrapper around STRMID (say STRMIDV) that  
> would have a simpler syntax for the case of single extractions from  
> multiple strings.

Aha! Where there's a will... Pavel, rejoin your faith.

A bit of a refinement, for the lazy among us:

```
new=strmid(key,0,transpose(p1))+'50'+strmid(key,tranpose(p2) )
```

The key is putting the threading vector on its head, as a column vector. Another simplification arrives from strmid's willingness to loop back over vectors which are too short (like the scalar 0), and to extract all the way to the end of a string, if no length is specified.

Row vectors are interpreted as multiple places in each given string to operate. Power, with a price.

Note that an even easier notation appears if you have p1 and p2 as columns in an array, e.g.:

```
IDL> p=[ [1,5], [2,6], [3,6], [4,8] ]
```

then you can simply use the relatively clean:

```
new=strmid(key,0,p[0,*])+'50'+strmid(key,p[1,*])
```

Here is a perfect case of where IDL's notion of keeping leading dimensions of size 1 is critical. Note that `p[0,*]` are lengths, and `p[1,*]` are subscripts.

This also works quite well for `p1` and `p2` not the same length as `key`, or each other.

JD

---