Subject: Re: Indexing problems under V6
Posted by JD Smith on Mon, 02 Apr 2007 21:48:30 GMT
View Forum Message <> Reply to Message

On Fri, 30 Mar 2007 10:41:54 +0100, Ian Dean wrote:

> Hi,
> This is almost a repeat of a psoting by Wayne Landsman, 12-Aug-2003.
> Because of contracts, we have been stuck on v5.5 for years!!!
> However, we are now in a position to upgrade to 6.3. As part of this, a
> great deal of our code is now broken because of the change in indexing.
> In 5.5,
> a={x:indgen(10)}
> b=[1, 4, 10]
> c=[4]
> val=a.x[c] <<<<<<<<<<this was a scalar of 4
> pos=where(b eq val) <<<<<<<<<<<<< returned [1]
>
> In 6.3.
> a={x:indgen(10)}
> b=[1, 4, 10]
> c=[4]
> val=a.x[c] <<<<<<<<<<this is now an array of [4]
> pos=where(b eq val) <<<<<<<<<<<<< now returns -1
>
> The where fails because it is now comparing an array with an array, not an
> array with a scalar.
> Is there any trick/switch that can be used to revert the action to that used
> in 5.5?
>
> The real problem is much deeper than this, as the search may be done several
> levels lower than where the selection criteria is setup and furthermore,
> some searches genuinely use arrays.
>
> I hope this makes sense and I hope for some favourable replies.

Probably not what you want to hear, but... it's important to know whether you intend the "scalar inflation" of IDL's operators to work, and, if so, always explicitly use a scalar, e.g. val[0]. This is a good habit to get into. Often I recommend turning an array into a scalar as soon as its returned, if you only expect, or ever intend to use, a single value:

```
wh=where(array eq val,cnt)
wh=wh[0]
```

Short of that, you could use WHERE_ARRAY in place of WHERE, at a cost of some performance (or write your own version to replace WHERE which

simply checks for single element arrays, and converts them to scalars for you). If it's any solace, I know that at least one of IDL's former engineers regarded the scalar <-> single element array ambiguity as the most troubling vestigial issue in IDL.

JD
