Subject: Re: Array slices
Posted by Anne[1] on Thu, 02 Feb 2006 16:53:10 GMT
View Forum Message <> Reply to Message

To convert everything to a [3,3] array just use the REFORM function, ie
arr=reform(arr). This will get rid of and redundent dimensions.
Actually I find it more annoying when the [3,3,1] array gets converted
to a [3,3] array since that is usually harder to deal with!
Anne
Richard Edgar wrote:

```
>  Greetings,
>
>  I'm having trouble with slices of 3D arrays in IDL, and I suspect that
>  it's something I don't understand about the way IDL interprets things.
>  My problem can be summed up as follows:
>
>  IDL> arr=DINDGEN(3,3,3)
>  IDL> help,arr
>  ARR             DOUBLE    = Array[3, 3, 3]
>  IDL> help,arr[*,*,1]
>  <Expression>    DOUBLE    = Array[3, 3]
>  IDL> help,arr[*,1,*]
>  <Expression>    DOUBLE    = Array[3, 1, 3]
>  IDL> help,arr[1,*,*]
>  <Expression>    DOUBLE    = Array[1, 3, 3]
>
>  My questions are:
>  a) Why don't all three slices return a [3,3] array?
>  b) How do I eliminate the 'useless' dimension?
>
>  I typically have routines which want a 2D array, which I only get from
>  the first of these. They (naturally) choke when given a 3D array, one of
>  whose dimensions is unity.
>
>  Thanks in advance,
>
>  Richard
```

Subject: Re: Array slices
Posted by David Fanning on Thu, 02 Feb 2006 16:58:03 GMT
View Forum Message <> Reply to Message

Richard Edgar writes:

```
>  I'm having trouble with slices of 3D arrays in IDL, and I suspect that
>  it's something I don't understand about the way IDL interprets things.
>  My problem can be summed up as follows:
```

```
>
> IDL> arr=DINDGEN(3,3,3)
> IDL> help,arr
> ARR             DOUBLE    = Array[3, 3, 3]
> IDL> help,arr[*,*,1]
> <Expression>    DOUBLE    = Array[3, 3]
> IDL> help,arr[*,1,*]
> <Expression>    DOUBLE    = Array[3, 1, 3]
> IDL> help,arr[1,*,*]
> <Expression>    DOUBLE    = Array[1, 3, 3]
>
```

> My questions are:
> a) Why don't all three slices return a [3,3] array?
> b) How do I eliminate the 'useless' dimension?
>
> I typically have routines which want a 2D array, which I only get from
> the first of these. They (naturally) choke when given a 3D array, one of
> whose dimensions is unity.

One man's "useless" dimension is another man's "essential"
dimension, as you will learn if you follow this newsgroup
for any length of time. The lamentations over the "missing"
dimension are legendary.

But, if it is really useless to you, then REFORM it away:

```
   IDL> Help, Reform(array[3,1,3], 3, 3)
 <Expression>    DOUBLE    = Array[3, 3]
```

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Array slices
Posted by Richard Edgar on Thu, 02 Feb 2006 17:52:36 GMT

View Forum Message <> Reply to Message

Anne wrote:

> To convert everything to a [3,3] array just use the REFORM function, ie
> arr=reform(arr). This will get rid of and redundent dimensions.
> Actually I find it more annoying when the [3,3,1] array gets converted
> to a [3,3] array since that is usually harder to deal with!

Thanks :-)

For me, the most annoying thing is the inconsistency. If all gave a
{3,3,1} array (permute as necessary), then there'd be less of a problem.
However to my mind, it would still be better to return [3,3] every time,
as happens in Fortran. For example, what if the next time I had a
[3,3,3,3] array, and wanted to hand off the 2D slice [1,*,2,*] for analysis?

Richard