
Subject: Re: Loop Arrays

Posted by [Ken Mankoff](#) on Mon, 15 Oct 2001 16:35:09 GMT

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

On Tue, 9 Oct 2001, Mark Hadfield wrote:

> From: "Ken Mankoff" <mankoff@lasp.colorado.edu>

>> I am interested in creating circular arrays, where subscripts that would
>> be out-of-bounds on a regular array just start indexing on the other side
>> of the array.

> BTW another indexing extension that appeals to me is Python's use of
> negative indices to refer to positions relative to the end of the array,
> eg -1 refers to the rightmost element.

I have a few ideas for implementing this feature. If you are interested in the option and ability to do this, please let me know your suggestions on the following.

The algorithm is simple: Evaluate the numbers, variables, and functions that make up the subscripts w.r.t. the dimensions of the arrays. If any are negative or larger than their dimension, then re-write code to behave in new way.

What is needed is a regular expression filter and replacement algorithm.

If any of the following is detected, then replace with:

```
a[ -3 ] a[ n_elements(a)-3 ]  
a[ 3-n ] if( 3-n lt 0 ) then a[ n_elements(a)-3-n ] else a[3-n]  
a[ 2, -3 ] a[ 2, (size(a,dim))[2] -3 ]  
a[ 2:-4 ] a[ 2: n_elements(a)-4 ]  
a( -3 ) * let normal interpreter handle ()  
etc...
```

I wrote this in IDL (for about 1/2 the cases) before I realized it has some pitfalls: `retail` and `.full_reset_session` exit the routine, and this routine tries to emulate `$MAIN$`, which is bad.

I realized I can fix those problemn with emacs lisp or Perl. With either one, I can open a pipe between IDL and the user, give them the ability turn on/off `wierd_subscript_method`, and then reformat their input so the IDL interpreter understands it. This could be done in any language, but I assume that Perl would be best since it is good with regexp and string manipulations. Does anyone have any other suggestions?

-k.
