Subject: Re: idl2matlab translate-o-matic
Posted by David McClain on Tue, 22 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

I knew that if you thought about it you would begin to understand my
point...

Try this one... Give me a function to return the array result of removing
selected items from an argument array.

Since I can't pass a testing function to that routine (IDL doesn't have
higher order functions), I will accept a routine, for illustrative purposes,
that removes all even values from the array.

Now suppose some joker passes an array containing only even values to that
routine...

- DM

Craig Markwardt <craigmnet@cow.physics.wisc.edu> wrote in message
news:onya8drxcf.fsf@cow.physics.wisc.edu...
>
> Pavel Romashkin <pavel@netsrv1.cmdl.noaa.gov> writes:
>
>>> What can you say of a language that is purely array oriented, but
>>> cannot comprehend the existence of an empty array?
>>
>> Agreeing with D.F., I so far had no use for an empty array. I
>> understand it is not flexible, but I usually work on data other than
>> nothing.
>
> Forgive him, he knows not what he says.
>
> Empty arrays would be invaluable in both indexing (such as with WHERE)
> and array concatenation. By invaluable, I mean that it would remove a
> lot of the special casing. Consider these examples:
>
> ARRAY INDEXING - indexing with where()
> *With* an empty array:
>   wh = where(array GT thresh, /EMPTY)
>   array(wh) = 0   ;; indexing with empty array has no effect
> *Without* an empty array
>   wh = where(array GT thresh, count)
>   if count GT 0 then array(wh) = 0
>
> ARRAY CONCATENATION - growing an array
> *With* an empty array:
>   l = empty_array()

```
>     for i = 0, 100 do if expression(values) then l = [l, values]
>   *Without* an empty array:
>     for i = 0, 100 do $
>       if expression then $
>          if n_elements(l) EQ 0 then l = [values] else l = [l, values]
>
>  As you can see, the "with" code is more simple and easy to read.  The
>  "without" (which represents the status quo) has special cases which
>  ruin the flow of thought.  For a vectorized language, this is a
>  painful burden to bear sometimes.  If you don't believe me, try doing
>  the following (apparently simple) problem:
>
>   * given two arrays, A and B: concatenate all but the last two
>     elements of A, with B.  Don't try [A(0:n-3),B], or you will be in a
>     world of hurt.
>
>  Craig
>
>  --
>   ------------------------------------------------------------ --------------
>  Craig B. Markwardt, Ph.D.         EMAIL:    craigmnet@cow.physics.wisc.edu
>  Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
>   ------------------------------------------------------------ --------------
```