Subject: IDL Array Push
Posted by Aram Panasenco on Sun, 04 Apr 2010 23:38:30 GMT
View Forum Message <> Reply to Message

I decided to write an IDL procedure an analogue of which is commonly
used in more modern computer languages: Array.push().

```
;+
; NAME:
;   PUSH
;
; PURPOSE:
;   Given two input arrays, the procedure modifies the first array by
"pushing" the
;   elements of the second array onto it. If both arrays are linear,
the elements of
;   the second array are added in-line: to the right of those of the
first array.
;   If any of the arrays is 2-dimensional, the elements of the second
array are added
;   below those of the first array.
```

Since this is the first data manipulation procedure I have ever
written for public use, I expect there to be some mistakes or errors I
have overlooked. I hope you will be willing to use it and inform me
about any bugs you find!

Feel free to use and distribute.

Pastebin link:
http://idl.pastebin.com/SQpSLvks

~Aram Panasenco

Subject: Re: IDL Array Push
Posted by JDS on Wed, 07 Apr 2010 21:04:40 GMT
View Forum Message <> Reply to Message

On Apr 6, 2:23 pm, Aram Panasenco <panasencoa...@gmail.com> wrote:
> On Apr 6, 3:52 am, Jeremy Bailin <astroco...@gmail.com> wrote:
>
>
>
>
>
>> On Apr 5, 1:36 pm, Aram Panasenco <panasencoa...@gmail.com> wrote:
>

>>> On Apr 5, 9:23 am, James <donje...@gmail.com> wrote:
>
>>>> How is this different from concatenating two arrays like [[[array1]],
>>>> [[array2]]]?
>
>>> *facedesk*
>
>>> It's not. I completely forgot about concatenating arrays. Oh well, I
>>> suppose it was good coding practice anyway.
>
>> In my version of PUSH, it concatenates the two arrays if they exist,
>> but creates the first one with the contents of the second if it
>> doesn't. That makes it easy to stick into a loop where you don't know
>> whether the original array will exist at entry (or if you explicitly
>> know that it won't).
>
>> -Jeremy.
>
> Wow, that's a great idea! It makes my code looks so much cleaner.
> Compare:

The only problem with this type of "push" is that it differ
dramatically from similar operators in other languages. For these,
typically extra "space" is pre-allocated at the array ends and used as
needed, intelligently adding more buffer as necessary. In principle
this could be done with IDL, but concatenation or other simple methods
make a full copy of the array each and every time an element is added,
which is costly.

JD