On Nov 10, 10:07 pm, Demitri wrote:
> Hello,
>
> I'm starting to dig into IDL programming, and one thing I've found
> particularly annoying is the lack of support for empty arrays. Although
> I'm using IDL's OO capabilities, I find that I'm spending a lot of time
> testing for return results where an empty array would be much more
> suitable.
>
> If I'm building an array that will have zero or more entries, I create
> a variable, "insert" a dummy value, go through the code adding entries
> based on whatever logic, then pop the top dummy value. But even then I
> have to check that more than one entry is there so I dont throw an
> error with a
>
> return, a[1:*]
>
> This is very basic functionality. Does anyone have a particularly
> elegant... well, I'd say solution, but anything is obviously a
> workaround. I don't want to fight the language, but I get the feeling
> that IDL's philosophy is "yeah, you do it yourself. I can't be
> bothered."
>
> Cheers,
>
> Demitri

Hi Demitri,

For these sorts of cases, I tend to use empty pointers. Here's an
example using pointers to provide the functionality I think you're
looking for:

var = ptr_new(/allocate_heap)
help, *var
IDL> <PtrHeapVar1>   UNDEFINED = <Undefined>

This article by JD Smith on David Fanning's website should help get
you started on pointers:
http://www.dfanning.com/misc_tips/pointers.html

Good luck!
Chris