

---

Subject: Re: exposing variables to higher program levels

Posted by [lasse](#) on Thu, 17 May 2007 11:08:57 GMT

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

---

On 17 May, 02:12, Christopher Thom <c...@oddjob.uchicago.edu> wrote:

> Hi all,  
>  
> I've been thinking for a while that I really need to kick my addition to  
> the IDL 'save' command...which, while very convenient, has the nasty habit  
> of not being as portable as I'd like. I recently got bitten by this  
> (again!).  
>  
> So this afternoon I started tooling around with two routines that would  
> dump all the variables I want into an anonymous structure, and save them  
> to a binary fits table. This works quite nicely, and I can then restore  
> the structure from the fits file, and extract the variables from the  
> structure.  
>  
> The one catch, though, is that I would like to wrap this up in two  
> routines, to do the bundling and unbundling from the needed  
> structure...but this seems to require the ability to export variables to a  
> higher program level. I don't think it's simply as easy as returning them  
> as parameters, since there are an arbitrary number of variables one might  
> restore from a given fits "save" file.  
>  
> Any ideas on how to do this (or code already written) would be greatly  
> appreciated.  
>  
> cheers  
> chris

Hi,

I have been reading in the help of SCOPE\_VARFETCH and under the ENTER keyword it says:

By default, SCOPE\_VARFETCH will return only variables that already exist in the specified scope. Attempts to access a nonexistent variable will cause IDL to issue an error and halt execution of the program. Set this keyword to alter this behavior. If ENTER is set and the desired variable does not exist in the specified scope, SCOPE\_VARFETCH will create a new variable within that scope and return the new variable. This keyword can be used to export data into other scopes. Generally, the scope of the calling routine and that of \$MAIN\$ are most likely to be useful destinations.

So I tried:

```
pro example2
  example
  print, haha
end
```

```
pro example
  void = scope_varfetch('haha', /enter, level=-1)
  void = 'Hello'
end
```

But alas!:

IDL> example2

IDL: Variable is undefined: <No name>.

IDL: Execution halted at: EXAMPLE 7 /home/lbnc1/idl/

example.pro

IDL: EXAMPLE2 2 /home/lbnc1/idl/

example.pro

IDL: \$MAIN\$

What am I doing wrong? Is that not exactly, what Chris wants?

Cheers

Lasse

---