Subject: Re: Looping over parameters without EXECUTE()
Posted by JD Smith on Tue, 03 May 2005 17:41:12 GMT
View Forum Message <> Reply to Message

On Tue, 03 May 2005 15:43:33 +0200, Thomas Pfaff wrote:

>
>
> JD Smith schrieb:
>> On Mon, 02 May 2005 12:10:43 -0400, Wayne Landsman wrote:
>>
>>
>>> The one case where I haven't figured out how to remove EXECUTE() from a
>>> program (to allow use with the Virtual Machine) is where one wants to
>>> loop over supplied parameters. For example, to apply the procedure
>>> 'myproc' to each supplied parameter (which may have different data
>>> types) one can use EXECUTE() to write each parameter to a temporary
>>> variable:

>
> How about putting all those parameters into a (named or anonymous)
> struct? Then you can have different types for each parameter and you're
> still able to loop over the elements.
>
> pro doit, param_struct
> for i=0, n_tags(param_struct) -1 do begin
> arg = param_struct.(i) ;->this way you can even store result values
> myproc, arg
> param_struct.(i) = arg
> endfor
> end
>
> Would that be a possibility, or am I missing something?

That works OK, and if you used TEMPORARY() you could cut down the data copying penalty. There are two main problems with this approach: 1) the user has to create a potentially large struct as input and then unpack it as output, which is not convenient from the command line, especially for output arguments, and 2) the type and size of each argument cannot be changed, thanks to the nature of structs.

JD