Subject: Re: call_procedure with a dynamically created arguments list?
Posted by Dominic Metzger on Fri, 06 Oct 2006 21:54:38 GMT
View Forum Message <> Reply to Message

Unfortunately, I do have to care about bicyclist with brain damages.
Since I am going to wrap IDL functions, I need to still return the same
kind of error message as when the IDL function is called directly. I
wish I wouldnt have to deal with invalid code but I have to.
All I am trying to do is to make the wrapper functions as transparent
to the user as possible. I dont care about avoiding errors if the user
enters invalid arguments or not the right number of arguments. I just
want to provide the user with the same error messages than he would
have gotten if he didnt use the wrapper functions.

dometz

P.S. "Groundhog Day"? That was a great movie though!

David Fanning wrote:
> Dometz writes:
>
>> I have already discussed this with a few of you but wanted to see if
>> anyone else might know a way of doing the following:
>>
>> Is it possible to do something like:
>>
>> foo = "bar"
>> bla = 'blu"
>> args=[]
>> if (N_ELEMENTS(foo) ne 0) then
>>   args=[args, foo]
>> if (N_ELEMENTS(bla) ne 0) then
>>   args=[args, bla]
>>
>> CALL_PROCEDURE('some_proc', args)
>>
>> So, the goal is to build dynamically an arguments list that will be
>> passed in to CALL_PROCEDURE, CALL_FUNCTION or CALL_METHOD. (I want to
>> avoid EXECUTE)
>
> Do you mean that your statistical sample of responses
> it too low to calculate a meaningful sample error?
> Or, do you want to know what happens when they guy
> who is going to use your program falls off his bike
> on the way to the office and hits his head and passes
> parameter two into the function as parameter one in his
> confusion?

>
> I have to admit, I am totally confused as to what
> you want from us. It's a five line program. We have
> covered just about every possibility here, from
> normal to malicious users. What is it you
> think we have missed?
>
> If you want an dynamical list of arguments, use
> a pointer array. But to what purpose? To avoid a
> possible error? Forget it. Your crazed bicyclist
> will screw you up every time.
>
> Cheers,
>
> David
>
> P.S. Maybe it's just me, but I feel like I'm on the
> set of Groundhog Day. :-(
>
>
>
>
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")