

---

Subject: Re: eval

Posted by [Dominic Metzger](#) on Wed, 04 Oct 2006 04:29:38 GMT

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

---

Hi David,

"but the rest of the parameters are passed normally,  
> as if to the correct procedure/function/method."

1) Is there a hack around this so that I could use call\_function,  
call\_procedure or call\_method dynamically?

2) Or at least build an array of the arguments so that I could do that  
alternatively? Like in Perl where you can pass in an array of  
arguments... hmmm, that might not work for name arguments though...

thanks,

dometz

PS: If I call a function with EXECUTE, is there a way to get the return  
value?

David Fanning wrote:

> Dometz writes:

>

>> Would this also be possible with call\_function or call\_procedure since

>> EXECUTE causes certain limitations and inefficiencies.

>>

>> Something like:

>> IDL> foo="bar, arg1, arg2, arg3, arg4=foo"

>> IDL> call\_procedure(foo)

>>

>> I want to dynamically build the list of arguments as a string and the

>> call the whole thing. Is this possible with call\_function or

>> call\_procedure?

>> Or is it possible to pass in an array containing the arguments

>> (including named args) somehow?

>

> Call\_Procedure, Call\_Method, and Call\_Function, along

> with Execute, I call "virtual" functions. With Execute

> you create the actual IDL command you want to execute

> as a string. With the other three virtual functions

> the name of the procedure/function/method is passed

> as a string (usually so it can be determined at run-time)

> but the rest of the parameters are passed normally,

> as if to the correct procedure/function/method.

>  
> Cheers,  
>  
> David  
> --  
> 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.")

---