Subject: Re: using of EXECUTE???
Posted by davidf on Fri, 08 Nov 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Hendrik Roepcke <alpha@jungle.toppoint.de) writes:

> some people of this group warned me [not?] to use
> the IDL option: EXECUTE("???")

Who in the world are you talking to Hendrik? Nobody
on this group would *ever* say such a thing!

> It will mess up the memory-management of IDL
> after a while they estimated...

They don't know what they are talking about. :-)

> But this option is wunderfull for dynamical
> management of data-array!!
>
> EXECUTE(name_of_array+"=findgen("+string(dim_of_array)+") ")

You bet!

> shall I use execute or not?

Sure, use it! It's powerful. It's fun. And it will prove to your
friends that you are a *real* IDL programmer!

> any hints?

Well, your friends probably used it to create variables in a loop
and had a lot of problems which caused them to warn you off.
Don't get so carried away with EXECUTE that you start using
it for everything. For example, used inside an IDL procedure
or function, you can't create variables or symbols that don't
already exist there. (Kind of like what you are doing above.)

If the variable "name_of_array" doesn't exist in your
IDL program where you are doing this EXECUTE, you
will not be able to execute this statement more
than 2-3 times before something bad happens.

You can fix your code, by doing something like this:

```
   name_of_array = 0
   ok=EXECUTE(name_of_array+"=findgen("+string(dim_of_array)+ ")")
```

Now you can execute the EXECUTE statement as many times as you like and you will never have any trouble with it. (At least none that might be described as "messing up memory management".)

By the way, I hear that the limitation on creating variables like this will be going away in the next version of IDL. And--I hope--so too will a limitation that the string you pass to EXECUTE can only have 131 characters in it.

Why 131? You don't want to know! And I couldn't tell you in any case. It's one of the weird things about IDL that causes me to love it so. (By the way, this is the secret password to the "IDL Expert Programmers" convention, held every year in Boulder. They ask the question: "How many characters in EXECUTE?" and if you answer "131" you get in. Please don't tell them you heard it from me.)

One other hint. EXECUTE and its cousins CALL_PROCEDURE and CALL_FUNCTION are dynamite in widget programs. *Especially* if you want re-sizeable graphics windows.

Have fun!

David

****
* David Fanning, Ph.D.
* 2642 Bradbury Court, Fort Collins, CO 80521
* Phone: 970-221-0438 Fax: 970-221-4762
* E-Mail: davidf@dfanning.com
*
* Sometimes I go about pitying myself, and all along my
* soul is being blown by great winds across the sky.
*   -- Ojibway saying
****

Subject: Re: using of EXECUTE???
Posted by Peter Mason on Mon, 11 Nov 1996 08:00:00 GMT

View Forum Message <> Reply to Message

On 8 Nov 1996, Hendrik Roepcke wrote:
> some people of this group warned me to use
> the IDL option:
> EXECUTE("???")
> It will mess up the memory-management of IDL
> after a while they estimated...

>
> But this option is wunderfull for dynamical
> management of data-array!!
> EXECUTE(name_of_array+"=findgen("+string(dim_of_array)+") ")
> shall I use execute or not?
> any hints?

I'd recommend using EXECUTE() only when there isn't a reasonably simple "conventional" way to achieve your goal; e.g., for evaluating expressions typed in by a user of your program.
The docs point out two issues:
. "Compiling the string (execute's arg) at run-time is inefficient..."
. "Do not use EXECUTE to create new variables inside procedures and functions... (it will fail)" (Actually, this does seem to work with a full IDL license. I suspect that it might be crippled under runtime IDL, though - imagine runtime IDL with this restriction lifted.)

In your example above, the only extra facility that EXECUTE gives you which you wouldn't get by simply doing SOME_VARNAME=FINDGEN(DIM_OF_ARRAY) is indirect specification of the array's name. I think that this is really a disadvantage - your program would always have to go via EXECUTE('some op referencing name_of_array') to USE this new array, as it would only know the array's name indirectly.

Peter Mason