
Subject: Re: Function call string
Posted by [David Fanning](#) on Fri, 10 Apr 2015 13:09:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Gordon Farquharson writes:

>
> Is there a way to access the string that was used to call a function or procedure in IDL? For example, lets say I call:
>
> display, 'A carrot is orange'
>
> Is there a way to access the string "display, 'A carrot is orange'" from within the display procedure?
>
> The reason I'd like to do this is that I'd like to record how a function or procedure was called in the meta data that the function or procedure generates, so that it is easy to make the same call again in the future.
>
> (I seem to recall that there was a discussion on this topic previously, but I can't find the thread.)
>
> I realize that I could construct the function call string from the arguments, but the output is not generic or pretty because one has to decide on things like how many decimal places to use to represent floating point values, and one has to take care of each keyword setting.
>
> Maybe a better self-documenting way to do this is to have a configuration file, e.g., an XML file that the procedure or function reads, and that accompanies the output. While I have done this previously for other routines, it seems a little overkill for my current application. Maybe it is the best solution though.
>
> (Not sure if David is still posting tips for his web page, but this would be a cool one to add if there is a neat solution. Maybe he already has it there.)

In a rudimentary way (not the elegant way you describe), this idea of "saving a command" is at the heart of the entire Coyote Graphics Library, since this is exactly what needs to be done to create a resizable graphics window with old graphics commands. See the `cgCmdWindow` object code for how it is done.

I couldn't figure out how to make the Coyote system completely generic to any and all commands with the "old" IDL commands I wanted to remain faithful to, so this works only within the limited universe of graphics commands. But, I think with the creative use of some of IDL's new language features, that something like this could now be written generically.

For example, one limitation in the Coyote Graphics System is the inability to create (or use) an output keyword. I read not too long ago

on the IDL Blog in the ExelisVis site about a convoluted way to make this possible with hashes. It's not simple, but it is probably possible.

I force the user to specify the "command" as a string. But, it is easy enough to get the name of the program module that called another. You can use `cgWhoCalledMe` from the Coyote Library, for example.

<http://www.idlcoyote.com/programs/cgwhocalledme.pro>

Cheers

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>

Sepore ma de ni thue. ("Perhaps thou speakest truth.")
