Subject: Re: ROUTINE_INFO problems
Posted by JD Smith on Fri, 12 Apr 2002 17:44:33 GMT
View Forum Message <> Reply to Message

Mark Hadfield wrote:
>
> "Ted Cary" <tedcary@yahoo.com> wrote in message
> news:3CB4C6E7.657CD4A5@yahoo.com...
>> Mark Hadfield wrote:
>>
>>> Try the ROUTINE_INFO function with keyword PARAMETERS set and
>>> examine the KW_ARGS tag in the structure it returns. (I found this
>>> in the docs & I haven't tried it myself.) Just out of curiosity,
>>> why do you want to do this?
>>
>> Thanks for the tip. ROUTINE_INFO sounded like exactly what I want,
>> but I played with it for a bit and it doesn't work as expected, at
>> least not for system routines like PLOT. In fact, I can't imagine
>> it does what anyone wants with system routines, since the PARAMETERS
>> structure it returns is just wrong.
>
> Well I *did* say I hadn't tried it myself.
>
>> ...Here is the output for the 'PLOT' procedure.
>>
>> IDL> plotParams = ROUTINE_INFO('PLOT', /PARAMETERS)
>> IDL> PRINT, plotParams.num_args
>>   0
>> IDL> PRINT, plotParams.num_kw_args
>>   0
>>
>> That's not right. I tried ROUTINE_INFO on PTR_FREE and even on
>> itself, but the results were the same. Probably this is all
>> documented or I'm just messing up, but why does ROUTINE_INFO return
>> the PARAMETERS structure for system routines if it's going to lie
>> about it?
>
> You don't really want an answer to that question, do you? Not from
> anyone other than the author of the routine, anyway. Though David
> could probably think up something pithy & relevant. To do with tennis,
> probably...
>
> I think the answer is "Just because".
>
>> The parameter information it returns for non-system routines also
>> could be more complete. If a routine uses keyword inheritance and
>> passes along an _EXTRA structure to a subroutine, then the KW_ARGS
>> field of the PARAMETERS structure returned by ROUTINE_INFO only

>> contains the word '_EXTRA.' It would be more useful to know *all*
>> the keywords that could be passed to the routine, including keywords
>> of any subroutines called with _EXTRA .
>
> I rather expected that one.
>
>> I want this information because I'm toying around with an idea that
>> will probably go nowhere. I'm trying to the use the keywords as
>> Get/Set-able properties of an object class, if that makes any sense.
>
> Not entirely, but it *would* be kind of cool to be able to query an
> object to see what keywords its GetProperty and SeProperty methods
> support.
>
>> So is there any way to ascertain all the keywords accepted by any
>> IDL routine, including keywords of system routines and including
>> keywords inherited from subroutines?
>
> Don't know, sorry (though I suspect not). JD is probably the expert on
> this (as on many other things) because there is a routine-info
> facility built into the IDLWAVE Emacs mode, which he currently
> maintains. But it can't recurse into inheritance chains either.

Not entirely true... newer versions of IDLWAVE do indeed follow
inheritance chains for keyword information in association with object
inheritance, but they don't do this by default for all routines with
*_EXTRA. Why? Looking through the body of the code for calls with
_EXTRA is error-prone and time consuming, and often returns something
other than what you want.

That said, you should know that IDLWAVE is very devious in the way it
gets information, relying on text versions of the PDF manuals for the
bulk of the system info. That's why it knows all the keywords to PLOT.

JD