Subject: Re: What subprogram? What parameters and keywords?
Posted by Moritz Fischer on Mon, 26 Aug 2013 13:05:32 GMT
View Forum Message <> Reply to Message

Hi again!

```
help, scope_varfetch('no_such_variable', LEVEL=-1, /ENTER)
<No name> UNDEFINED = <Undefined>
```

or, to check this on runtime,

```
defined_flag = n_elements(scope_varfetch('no_such_variable', LEVEL=-1,
/ENTER))  NE 0
```

cheers

Am 26.08.2013 14:09, schrieb Mats Löfdahl:
> Den fredagen den 23:e augusti 2013 kl. 17:09:40 UTC+2 skrev Mats
> Löfdahl:
>> Den fredagen den 23:e augusti 2013 kl. 15:57:32 UTC+2 skrev Mats
>> Löfdahl:
>>
>>> Den fredagen den 23:e augusti 2013 kl. 14:52:28 UTC+2 skrev David
>>> Fanning:
>>
>>>> Mats Löfdahl writes:
>>
>>>
>>
>>>> > Let's say IDL has just entered a subroutine or function. Is
>>>> > any of the following information available?
>>
>>>
>>
>>>> > 1. The name of the subprogram (like $0 in bash)
>>
>>>> > 2. The parameters as an array or struct or similar (kind of
>>>> > like $*), preferably including keywords.
>>
>>>
>>
>>>> You can find two useful programs in the Coyote Library. WhoAmI
>>>> returns
>>
>>>> the name of the program module the WhoAmI program is called in,
>>>> and
>>

>>>> WhoCalledMe identifies the name of the program module that
>>>> called the
>>
>>>> routine you are now in.
>>
>>>
>>
>>>> There is no parameter stack that I am aware of, although it
>>>> wouldn't
>>
>>>> surprise me if the Scope_*** routines could figure it out.
>>
>>>
>>
>>> Thanks, that solves the first part of the problem! I'll read up
>>> on scope_*** and see if I can figure something out for the second
>>> part.
>>
>>
>>
>> I didn't give much detail in my original post, but what I'm trying
>> to do is make a subroutine that writes a log file with as much info
>> as possible about the subprogram it was called from, without having
>> been explicitly told very much. So David's WhoCalledMe is actually
>> better than what I asked for.
>>
>>
>>
>> And reading up on scope_*** and related routines was good advice,
>> too. Looks like I could do something like this:
>>
>>
>>
>> * Get the subprogram name from name=WhoCalledMe()
>>
>> * Get the names of parameters and keywords from
>> routine_info(name,/parameters)
>>
>> * Get the level of the calling function from scope_level()-1
>>
>> * Get the current values of the parameters and keywords from
>> scope_varfetch
>
> Another problem:
>
> The list of parameters I get from routine_info(name,/parameters)
> includes also those parameters that were not used in the call and
> therefore not defined. When I try to access the values of undefined

> variables, scope_varfetch throws an error.
>
> Is there a way to get the list of used parameters only, or a way to
> avoid the error when calling scope_varfetch with the name of a
> variable that is undefined?
>
>
> Is there a way to
>