
Subject: get variables name

Posted by [R.Bauer](#) on Wed, 05 Apr 2000 07:00:00 GMT

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

Hi

I like to know the name of a variable which is submitted as input to a procedure

e.g

```
pro test,var  
help,var  
end
```

```
a=1  
test,a
```

Is it possible to get inside test the information that test was called with a.

R.Bauer

Subject: Re: get variables name

Posted by [R.Bauer](#) on Sun, 09 Apr 2000 07:00:00 GMT

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

"J.D. Smith" wrote:

>

> David Fanning wrote:

>>

>> R.Bauer (R.Bauer@fz-juelich.de) writes:

>>

>>> I like to know the name of a variable which is submitted as input to a
>>> procedure

>>>

>>> e.g

>>>

>>> pro test,var

>>> help,var

>>> end

>>>

>>> a=1

>>> test,a

>>>

>>> Is it possible to get inside test the information that test was called

>>> with a.

```
>>
>> My life is totally out of control at the moment, so I haven't
>> had the opportunity to write this up for my web page, but
>> here is part of a correspondence I recently received from
>> Paul Woodford, an IDL user, concerning the undocumented
>> ROUTINE_NAMES function:
>>
>> routine_names(variables=0) - variable names at current level
>> routine_names(variables=1) - variable names at main level
>> routine_names(variables=2) - variable names at one level down from main
>> routine_names(variables=-1) - variable names at one level up from current
>> routine_names(variables=-2) - variable names at two levels up from current
>>
>> But wait, that's not all... Take a look at how routine_names is used in
>> gethelp - you can use routine_names to actually fetch a variable from a
>> different level. For instance, if you have a variable named 'blah' one
>> level up from your current level, you can get it using
>>
>> blah_here = routine_names('blah', fetch=-1)
>>
> < wise cautionary matter clipped >
>
> But wait... that's still not all!
>
> By poking around in the library, I found:
>
> pro foo,a
>   print,routine_names(a,/ARG_NAME)
> end
>
```

That's what I need.

I am using it now in this way.

```
clevel = ROUTINE_NAMES(/LEVEL)
in_name=routine_names(data,ARG_NAME=clevel-1)
```

R.Bauer
