Subject: Re: debugging with new variables (dictionary, hash, ...)
Posted by Helder Marchetto on Wed, 14 Jan 2015 22:35:20 GMT
View Forum Message <> Reply to Message

I can only offer a "happy birthday". I have no chance to access the code.
I think that the following are the same (at least in my default workbench configuration):
Keyboard = workbench = command line
F5 = "In" = ???
F6 = "Over" = .SO
F7 = "Out" = .OUT

There is also the "don't try this at home" option.
While in the dictionary or hash procedure, try what happens when you press Ctrl+Shift+F6 (=Skip). You will generate some "unexpected" errors like:

% SCOPE_VARFETCH: String expression required in this context: KEY.

Cheers,
Helder

On Wednesday, January 14, 2015 at 11:25:16 PM UTC+1, Jim P wrote:
> On Wednesday, January 14, 2015 at 3:19:10 PM UTC-7, wlandsman wrote:
>> You could use the .STEPOVER (.SO) command. ("Unlike .STEP, if .STEPOVER executes a statement that calls another routine, the called routine runs until it ends before control returns to interactive mode.")
>>
>> Or somewhat more tedious you could use .OUT once you enter into HASH or DICTIONARY to get out in one step.
>>
>> Now you just have to assign .SO to a key (which I assume can be done but don't know how offhand).
>>
>> --Wayne
>>
>>
>> On Wednesday, January 14, 2015 at 5:06:04 PM UTC-5, Helder wrote:
>>> Hi,
>>> I like to debug inserting breakpoints and then I like to use F5 or F6 (step-in and step-over) to go through the code.
>>> If you have some of the new variables in the code, it will be a painful clicking exercise...
>>>
>>> Here is a minimal example, that simulates what happens to me quite often...
>>>
>>> function retValue, inVar1, inVar2
>>> return, {inVar1:inVar1, inVar2:inVar2}
>>> end
>>>
>>> pro testDict

```
>>>  d = dictionary('var', 5,'new',6)
>>>  print, retValue(d.var, d.new) ;put a breakpoint on this line
>>>  end
>>>
```

>>> Now run testDict and then suppose you would like to go in the function "retValue" by clicking F5 (step-in). Well, good luck. It's going to take, in this example with two dictionaries, about 93 clicks of F5. Below is the terminal output.
>>>
>>> Is there any chance to see all of this disappear in the future?
>>>
>>> Regards,
>>> Helder
>>>
>>> PS: There are of course ways around this, like putting a new toggle point at the beginning of the procedure/function being called. However, the behavior does not fit with other IDL functions.
>>>

```
>>>  % Stepped to: DICTIONARY::GETPROPERTY   13
>>>  % Stepped to: DICTIONARY::GETPROPERTY   16
>>>  % Stepped to: DICTIONARY::GETPROPERTY   18
>>>  % Stepped to: DICTIONARY::GETPROPERTY   19
>>>  % Stepped to: DICTIONARY::HASKEY   91
>>>  % Stepped to: DICTIONARY::HASKEY   93
>>>  % Stepped to: DICTIONARY::HASKEY   96
>>>  % Stepped to: DICTIONARY::HASKEY   99
>>>  % Stepped to: HASH::HASKEY     428
>>>  % Stepped to: HASH::HASKEY     430
>>>  % Stepped to: HASH::HASKEY     432
>>>  % Stepped to: HASH::HASKEY     435
>>>  % Stepped to: HASH::HASKEY     437
>>>  % Stepped to: HASH::HASKEY     440
>>>  % Stepped to: HASH::HASKEY     442
>>>  % Stepped to: HASH::HASKEY     443
>>>  % Stepped to: HASH::HASKEY     444
>>>  % Stepped to: HASH::HASKEY     445
>>>  % Stepped to: HASH::HASKEY     448
>>>  % Stepped to: HASH::HASKEY     449
>>>  % Stepped to: HASH::HASKEY     451
>>>  % Stepped to: DICTIONARY::GETPROPERTY   20
>>>  % Stepped to: DICTIONARY::GET    77
>>>  % Stepped to: DICTIONARY::GET    79
>>>  % Stepped to: DICTIONARY::GET    82
>>>  % Stepped to: DICTIONARY::GET    85
>>>  % Stepped to: HASH::GET        213
>>>  % Stepped to: HASH::GET        216
>>>  % Stepped to: HASH::GET        218
>>>  % Stepped to: HASH::GET        221
>>>  % Stepped to: HASH::GET        223
>>>  % Stepped to: HASH::GET        226
>>>  % Stepped to: HASH::GET       227
>>>  % Stepped to: HASH::GET       231
>>>  % Stepped to: HASH::GET       232
>>>  % Stepped to: HASH::GET       233
>>>  % Stepped to: HASH::GET       236
>>>  % Stepped to: HASH::GET       237
>>>  % Stepped to: HASH::GET       238
>>>  % Stepped to: HASH::GET       243
>>>  % Stepped to: HASH::GET       246
>>>  % Stepped to: HASH::GET       247
>>>  % Stepped to: HASH::GET       255
>>>  % Stepped to: HASH::GET       256
>>>  % Stepped to: HASH::GET       261
>>>  % Stepped to: HASH::GET       265
>>>  % Stepped to: DICTIONARY::GETPROPERTY   37
>>>  % Stepped to: DICTIONARY::GETPROPERTY   13
>>>  % Stepped to: DICTIONARY::GETPROPERTY   16
>>>  % Stepped to: DICTIONARY::GETPROPERTY   18
>>>  % Stepped to: DICTIONARY::GETPROPERTY   19
>>>  % Stepped to: DICTIONARY::HASKEY   91
>>>  % Stepped to: DICTIONARY::HASKEY   93
>>>  % Stepped to: DICTIONARY::HASKEY   96
>>>  % Stepped to: DICTIONARY::HASKEY   99
>>>  % Stepped to: HASH::HASKEY     428
>>>  % Stepped to: HASH::HASKEY     430
>>>  % Stepped to: HASH::HASKEY     432
>>>  % Stepped to: HASH::HASKEY     435
>>>  % Stepped to: HASH::HASKEY     437
>>>  % Stepped to: HASH::HASKEY     440
>>>  % Stepped to: HASH::HASKEY     442
>>>  % Stepped to: HASH::HASKEY     443
>>>  % Stepped to: HASH::HASKEY     444
>>>  % Stepped to: HASH::HASKEY     445
>>>  % Stepped to: HASH::HASKEY     448
>>>  % Stepped to: HASH::HASKEY     449
>>>  % Stepped to: HASH::HASKEY     451
>>>  % Stepped to: DICTIONARY::GETPROPERTY   20
>>>  % Stepped to: DICTIONARY::GET    77
>>>  % Stepped to: DICTIONARY::GET    79
>>>  % Stepped to: DICTIONARY::GET    82
>>>  % Stepped to: DICTIONARY::GET    85
>>>  % Stepped to: HASH::GET       213
>>>  % Stepped to: HASH::GET       216
>>>  % Stepped to: HASH::GET       218
>>>  % Stepped to: HASH::GET       221
>>>  % Stepped to: HASH::GET       223
>>>  % Stepped to: HASH::GET       226
>>>  % Stepped to: HASH::GET       227
>>>  % Stepped to: HASH::GET        231
>>>  % Stepped to: HASH::GET        232
>>>  % Stepped to: HASH::GET        233
>>>  % Stepped to: HASH::GET        236
>>>  % Stepped to: HASH::GET        237
>>>  % Stepped to: HASH::GET        238
>>>  % Stepped to: HASH::GET        243
>>>  % Stepped to: HASH::GET        246
>>>  % Stepped to: HASH::GET        247
>>>  % Stepped to: HASH::GET        255
>>>  % Stepped to: HASH::GET        256
>>>  % Stepped to: HASH::GET        261
>>>  % Stepped to: HASH::GET        265
>>>  % Stepped to: DICTIONARY::GETPROPERTY   37
```

>

> There's also a ".out" button that was added to the workbench toolbar in 8.1.

>

> It would be nice if these classes could be shoved down to C code or the IDL source routines could be exposed instead of being hidden in a .sav file. If anyone can't think of something to get me for my birthday, this would do the trick.

>

> Jim P.