On Monday, 8 June 2015 07:49:14 UTC-7, Fabien wrote:
> On 06/08/2015 04:15 PM, Helder wrote:
>> Please save everything before trying it!
>>
>> IDL> a = dictionary('b',dictionary('c',findgen(10)))
>> IDL> a.b['c',1:-1]
>>
>> ... bye bye IDL
>
> Nice! Crashes IDL on my 64b Linux machine, too.
>
> Cheers,
>
> Fabien

I found some slightly different results:

IDL> !version
{
    "ARCH": "x86_64",
    "OS": "Win32",
    "OS_FAMILY": "Windows",
    "OS_NAME": "MicrosoftWindows",
    "RELEASE": "8.4",
    "BUILD_DATE": "Sep272014",
    "MEMORY_BITS": 64,
    "FILE_OFFSET_BITS": 64
}

IDL> a = dictionary('b',dictionary('c',findgen(10)))

IDL> a.b['c',[1:-1]]
      1.0000000   0.00000000   0.00000000
;; Mine does not fail like yours, but it seems wrong!
;; That looks like array indexing off the low end, filling in '0' as it reaches back to -1

;; This seems to get to what you want, I think!:
IDL> a.b.c[1:-1]
      1.0000000   2.0000000   3.0000000   4.0000000   5.0000000   6.0000000
      7.0000000   8.0000000   9.0000000
;; But maybe you want the 'c' to be a string, to change it programmatically, in which case:
IDL> (a.b['c'])[1:-1]
      1.0000000   2.0000000   3.0000000   4.0000000   5.0000000   6.0000000
      7.0000000   8.0000000   9.0000000

;; I believe the parentheses cause a temporary copy of the array(10) to be made, which may not be OK in your application

;; I wondered about modifying values in there, doesn't look good:
IDL> a.b.c[1:-1]=42
% Attempt to store into an expression: Structure reference.
% Execution halted at: $MAIN$
IDL> help,a.b.c[1:-1]
<Expression>   FLOAT   = Array[9]

;; I would think that should work, too:
IDL> c=findgen(10)
IDL> c[1:-1]=42
IDL> c
     0.00000000   42.000000   42.000000   42.000000   42.000000   42.000000
      42.000000   42.000000   42.000000   42.000000

Is anyone from {Harr|Exel}|is taking notes from this? Chris, are you there? :-)

Cheers,
-Dick

Dick Jackson Software Consulting Inc.
Victoria, BC, Canada --- http://www.d-jackson.com