Subject: buggy dictionary
Posted by Helder Marchetto on Mon, 08 Jun 2015 14:08:28 GMT
View Forum Message <> Reply to Message

Hi,
I've just encountered another bug using dictionaries.

This is ok:

```
IDL> a = dictionary('b',findgen(10))
IDL> print, a['b',1:-1]
      1.00000      2.00000      3.00000      4.00000      5.00000      6.00000      7.00000      8.00000
 9.00000
```

Doing the same, but inside an object (that being self.a['b',1:-1]) will make the IDLDE crash "hard" (=window closes without warning).
Here is an example:

```
pro testObjBug::testIt
print, self.aa['b',1:-1]   ;this makes IDL crash
end

function testObjBug::init
self.aa = dictionary('b',findgen(10))
return, 1
end

pro testObjBug__define
class = {testObjBug,$
        aa:obj_new()}
end

pro testObjBug
o = obj_new('testObjBug')
o->testIt
end
```

The only way around this is to copy the variable out of the object:

```
aa = self.aa['b']
print, aa[2:-2]
```

Cheers,
Helder

PS: I've submitted this to IDL. It's logged with incident nr 294397.

Subject: Re: buggy dictionary

Posted by Helder Marchetto on Mon, 08 Jun 2015 14:15:49 GMT
View Forum Message <> Reply to Message

On Monday, June 8, 2015 at 4:08:30 PM UTC+2, Helder wrote:
> Hi,
> I've just encountered another bug using dictionaries.
>
> This is ok:
> IDL> a = dictionary('b',findgen(10))
> IDL> print, a['b',1:-1]
> 1.00000 2.00000 3.00000 4.00000 5.00000 6.00000 7.00000 8.00000
 9.00000
>
> Doing the same, but inside an object (that being self.a['b',1:-1]) will make the IDLDE crash "hard" (=window closes without warning).
> Here is an example:
>
> pro testObjBug::testIt
> print, self.aa['b',1:-1] ;this makes IDL crash
> end
>
> function testObjBug::init
> self.aa = dictionary('b',findgen(10))
> return, 1
> end
>
> pro testObjBug__define
> class = {testObjBug,$
> aa:obj_new()}
> end
>
> pro testObjBug
> o = obj_new('testObjBug')
> o->testIt
> end
>
> The only way around this is to copy the variable out of the object:
> aa = self.aa['b']
> print, aa[2:-2]
>
> Cheers,
> Helder
>
> PS: I've submitted this to IDL. It's logged with incident nr 294397.

Sorry, but I just managed to make a much shorter example of this bug. Please save everything before trying it!

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

IDL> a.b['c',1:-1]

... bye bye IDL

Notice that this works:
IDL> a.b['c',1:9]
      1.0000000   2.0000000   3.0000000   4.0000000   5.0000000   6.0000000
7.0000000   8.0000000   9.0000000

But anything with the '-' after the ':' doesn't.

Cheers,
Helder

Subject: Re: buggy dictionary
Posted by Fabzi on Mon, 08 Jun 2015 14:49:11 GMT
View Forum Message <> Reply to Message

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

Subject: Re: buggy dictionary
Posted by dg86 on Mon, 08 Jun 2015 15:12:11 GMT
View Forum Message <> Reply to Message

On Monday, June 8, 2015 at 10:49:14 AM UTC-4, 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

Same is true for MacOS. Attempting a slice with a negative index gives the correct answer, but leaves IDL churning. After a few seconds, IDL crashes with Segmentation fault: 11

Cool!

Subject: Re: buggy dictionary
Posted by Helder Marchetto on Mon, 08 Jun 2015 15:22:22 GMT
View Forum Message <> Reply to Message

> Cool!

Not sure I find that cool. I can only tell you that if this syntax is hidden somewhere in a n-thousand line code, it takes you quite a few "F6" to get to find the error. Need soon a new keyboard :-)
Helder

Subject: Re: buggy dictionary
Posted by dg86 on Mon, 08 Jun 2015 16:28:48 GMT
View Forum Message <> Reply to Message

On Monday, June 8, 2015 at 11:22:23 AM UTC-4, Helder wrote:
>> Cool!
>
> Not sure I find that cool. I can only tell you that if this syntax is hidden somewhere in a n-thousand line code, it takes you quite a few "F6" to get to find the error. Need soon a new keyboard :-)
> Helder

I didn't mean to minimize your pain, just to compliment you on uncovering such a deep and subtle bug.

All the best,

David

Subject: Re: buggy dictionary
Posted by Dick Jackson on Mon, 08 Jun 2015 18:28:56 GMT

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

Subject: Re: buggy dictionary
Posted by Helder Marchetto on Mon, 08 Jun 2015 20:30:33 GMT
View Forum Message <> Reply to Message

On Monday, June 8, 2015 at 8:28:59 PM UTC+2, Dick Jackson wrote:
> 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

Hi,
I got some feedback from Exelis, but it's way past working hours so I will sum up tomorrow.
Dick, you're right. There is a bug.

Cheers,
Helder

Subject: Re: buggy dictionary
Posted by Helder Marchetto on Tue, 09 Jun 2015 09:32:29 GMT
View Forum Message <> Reply to Message

On Monday, June 8, 2015 at 8:28:59 PM UTC+2, Dick Jackson wrote:
> 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

Hi Dick,
you're absolutely right. There is a syntax error that causes the crash.
From IDL support I learned that I should have used the following syntax to index the array::

```
a = dictionary('b',dictionary('c',findgen(10)))
print, a.b['c',[1:-1]]
```

However, as you pointed out, this delivers the wrong answer. Exelis acknowledged this (incident number 294397) and filed a bug report.

So this will give a wrong answer:
print, a.b['c',[1:-1]]
but this will be ok:

print, a.b.c[1:-1]

Cheers,
Helder

PS: Funny extra. According to the dictionary access documentation (http://www.exelisvis.com/docs/DICTIONARY.html#Access) in the example at the very button there is a code example:

```
 ------------------------------------------------------------
str = {data: FINDGEN(10)}
PRINT, str.data
PRINT, str.data[2:5] ; this works

dict = DICTIONARY('data', FINDGEN(10))
PRINT, dict.data ; this works
PRINT, dict.data[2:5] ; this will fail with an error
PRINT, dict['data', [2:5]] ; this works correctly
------------------------------------------------------------
```

The line commented "this will fail with an error" does not fail at all.

```
IDL> dict = DICTIONARY('data', FINDGEN(10))
IDL> PRINT, dict.data ; this works
IDL> PRINT, dict.data[2:5] ; this will fail with an error
IDL> PRINT, dict['data', [2:5]] ; this works correctly
     0.000000      1.00000      2.00000      3.00000      4.00000      5.00000      6.00000      7.00000
  8.00000      9.00000
      2.00000      3.00000      4.00000      5.00000
      2.00000      3.00000      4.00000      5.00000
IDL> !version
{
   "ARCH": "x86_64",
   "OS": "Win32",
   "OS_FAMILY": "Windows",
   "OS_NAME": "Microsoft Windows",
   "RELEASE": "8.4.1",
   "BUILD_DATE": "Feb 17 2015",
   "MEMORY_BITS": 64,
   "FILE_OFFSET_BITS": 64
}
```

Subject: Re: buggy dictionary
Posted by Lajos Foldy on Tue, 09 Jun 2015 11:01:41 GMT
View Forum Message <> Reply to Message

On Tuesday, June 9, 2015 at 11:32:30 AM UTC+2, Helder wrote:
>

> a = dictionary('b',dictionary('c',findgen(10)))
> print, a.b['c',[1:-1]]
>
> However, as you pointed out, this delivers the wrong answer. Exelis acknowledged this (incident number 294397) and filed a bug report.
>
> So this will give a wrong answer:
> print, a.b['c',[1:-1]]

Here [1:-1] is not a subscript range, it is the colon operator for array creation (introduced in IDL 8.3):

```
IDL> x=[1:-1]
IDL> help, x & print, x
X               INT       = Array[3]
      1       0      -1
```

So [1:-1] is actually an array subscript (with the usual array subscript clipping).

> but this will be ok:
> print, a.b.c[1:-1]

Here [1:-1] is a subscript range with a negative subscript value.

So the two [1:-1]'s have different meaning.

regards,
Lajos

Subject: Re: buggy dictionary
Posted by Dick Jackson on Tue, 09 Jun 2015 16:10:46 GMT
View Forum Message <> Reply to Message

On Tuesday, 9 June 2015 04:01:43 UTC-7, fawltyl...@gmail.com wrote:
> On Tuesday, June 9, 2015 at 11:32:30 AM UTC+2, Helder wrote:
>>
>> a = dictionary('b',dictionary('c',findgen(10)))
>> print, a.b['c',[1:-1]]
>>
>> However, as you pointed out, this delivers the wrong answer. Exelis acknowledged this (incident number 294397) and filed a bug report.
>>
>> So this will give a wrong answer:
>> print, a.b['c',[1:-1]]
>
> Here [1:-1] is not a subscript range, it is the colon operator for array creation (introduced in IDL 8.3):

>
> IDL> x=[1:-1]
> IDL> help, x & print, x
> X   INT   = Array[3]
>   1   0   -1
>
> So [1:-1] is actually an array subscript (with the usual array subscript clipping).
>
>> but this will be ok:
>> print, a.b.c[1:-1]
>
> Here [1:-1] is a subscript range with a negative subscript value.
>
> So the two [1:-1]'s have different meaning.
>
> regards,
> Lajos

Oops, you're right! And I do find the same behaviour, when I actually test Helder's case:

```
IDL> a = dictionary('b',dictionary('c',findgen(10)))
IDL> a.b.c[[1:-1]]
      1.0000000      0.00000000      0.00000000
IDL> a.b.c[1:-1]
      1.0000000       2.0000000       3.0000000       4.0000000       5.0000000       6.0000000
      7.0000000       8.0000000       9.0000000
IDL> a.b['c',[1:-1]]
      1.0000000      0.00000000      0.00000000
IDL> a.b['c',1:-1]
      1.0000000       2.0000000       3.0000000       4.0000000       5.0000000       6.0000000
      7.0000000       8.0000000       9.0000000
```

Sorry for the miscommunication!

Cheers,
-Dick

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

Subject: Re: buggy dictionary
Posted by Dick Jackson on Tue, 09 Jun 2015 16:15:16 GMT
View Forum Message <> Reply to Message

On Tuesday, 9 June 2015 09:10:48 UTC-7, Dick Jackson wrote:

> Oops, you're right! And I do find the same behaviour, when I actually test Helder's case:

>
> IDL> a = dictionary('b',dictionary('c',findgen(10)))
> IDL> a.b.c[[1:-1]]
>   1.0000000   0.00000000   0.00000000
> IDL> a.b.c[1:-1]
>   1.0000000   2.0000000   3.0000000   4.0000000   5.0000000   6.0000000
>   7.0000000   8.0000000   9.0000000
> IDL> a.b['c',[1:-1]]
>   1.0000000   0.00000000   0.00000000
> 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 should have added:
... running this at a Windows command prompt, right after printing this correct answer, the hard crash happens!

Cheers,
-Dick

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

Subject: Re: buggy dictionary
Posted by chris_torrence@NOSPAM on Tue, 09 Jun 2015 18:21:11 GMT
View Forum Message <> Reply to Message

Hi all,

I've fixed this bug for IDL 8.5, and I've also updated the example in the dictionary docs. Thanks to everyone for tracking it down.

Cheers,
Chris
