
Subject: indexing a structure in a list

Posted by [Helder Marchetto](#) on Wed, 05 Feb 2014 16:36:09 GMT

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

Hi,

I've got the feeling this should work, but for some reason it isn't.

So here is some code to test the problem with the errors describe in it:

```
PRO testStrutRef
Compile_Opt idl2
a = list(findgen(10), /extract)
a[4] = {value:3.0}
help, a[4], /struct
;a[4].value = 4.0      ;won't work: % Attempt to store into an expression: Structure reference.
;(a[4]).value = 4.0   ;won't work: % Attempt to store into an expression: Structure reference.
;((a[4]).value = 4.0  ;won't work: % Attempt to store into an expression: Structure reference.
struct = a[4]
struct.value = 4.0
a[4] = struct        ;works, but not really straight forward...
print, a
END
```

It seems like I can't address a structure element of a list directly.

Did I misplace my parenthesis? Is there a less cumbersome workaround?

Thanks,
Helder

Subject: Re: indexing a structure in a list

Posted by [David Fanning](#) on Wed, 05 Feb 2014 17:00:44 GMT

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

Helder writes:

```
> I've got the feeling this should work, but for some reason it isn't.
> So here is some code to test the problem with the errors describe in it:
>
> PRO testStrutRef
> Compile_Opt idl2
> a = list(findgen(10), /extract)
> a[4] = {value:3.0}
> help, a[4], /struct
> ;a[4].value = 4.0      ;won't work: % Attempt to store into an expression: Structure reference.
> ;(a[4]).value = 4.0   ;won't work: % Attempt to store into an expression: Structure reference.
> ;((a[4]).value = 4.0  ;won't work: % Attempt to store into an expression: Structure reference.
> struct = a[4]
> struct.value = 4.0
```

```
> a[4] = struct      ;works, but not really straight forward...
> print, a
> END
>
> It seems like I can't address a structure element of a list directly.
> Did I misplace my parenthesis? Is there a less cumbersome workaround?
```

Yeah, don't know. I'm no expert, but I'm not surprised lists and structures don't play well together. Have you thought about a hash instead?

```
IDL> a = list(findgen(10), /extract)
IDL> a[4] = hash('value', 3.0)
IDL> (a[4])['value'] = 4.0
IDL> print, a[4]
value:    4.00000
```

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: indexing a structure in a list
Posted by [Craig Markwardt](#) on Wed, 05 Feb 2014 17:52:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wednesday, February 5, 2014 11:36:09 AM UTC-5, Helder wrote:

```
> Hi,
>
> I've got the feeling this should work, but for some reason it isn't.
>
> So here is some code to test the problem with the errors describe in it:
>
>
> PRO testStrutRef
> Compile_Opt idl2
> a = list(findgen(10), /extract)
> a[4] = {value:3.0}
> help, a[4], /struct
> ;a[4].value = 4.0      ;won't work: % Attempt to store into an expression: >
> ;(a[4]).value = 4.0   ;won't work: % Attempt to store into an expression: Structure reference.
```

```
> ;((a[4]).value = 4.0 ;won't work: % Attempt to store into an expression: Structure reference.
> struct = a[4]
> struct.value = 4.0
> a[4] = struct ;works, but not really straight forward...
> print, a
> END
>
> It seems like I can't address a structure element of a list directly.
> Did I misplace my parenthesis? Is there a less cumbersome workaround?
```

Seems like a flat-out bug to me. List elements should be "lvalues," as they say in the compile business.

Craig

Subject: Re: indexing a structure in a list
Posted by [Michael Galloy](#) on Wed, 05 Feb 2014 18:05:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 2/5/14, 10:52 AM, Craig Markwardt wrote:

```
> On Wednesday, February 5, 2014 11:36:09 AM UTC-5, Helder wrote:
>> Hi,
>>
>> I've got the feeling this should work, but for some reason it isn't.
>>
>> So here is some code to test the problem with the errors describe in it:
>>
>>
>> PRO testStrutRef
>> Compile_Opt idl2
>> a = list(findgen(10), /extract)
>> a[4] = {value:3.0}
>> help, a[4], /struct
>> ;a[4].value = 4.0 ;won't work: % Attempt to store into an expression: >
>> ;(a[4]).value = 4.0 ;won't work: % Attempt to store into an expression: Structure reference.
>> ;((a[4]).value = 4.0 ;won't work: % Attempt to store into an expression: Structure reference.
>> struct = a[4]
>> struct.value = 4.0
>> a[4] = struct ;works, but not really straight forward...
>> print, a
>> END
>>
>> It seems like I can't address a structure element of a list directly.
>> Did I misplace my parenthesis? Is there a less cumbersome workaround?
>
> Seems like a flat-out bug to me. List elements should be "lvalues," as they say in the compile business.
```

> Craig
>

Yes, seems like a bug. Note that index notation works:

```
IDL> print, (a[4]).(0)
      3.00000
```

Mike

--

Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (<http://modernidl.idldev.com>)
Research Mathematician
Tech-X Corporation

Subject: Re: indexing a structure in a list
Posted by [Helder Marchetto](#) on Wed, 05 Feb 2014 19:42:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wednesday, February 5, 2014 7:05:02 PM UTC+1, Mike Galloy wrote:

> On 2/5/14, 10:52 AM, Craig Markwardt wrote:

>

>> On Wednesday, February 5, 2014 11:36:09 AM UTC-5, Helder wrote:

>

>>> Hi,

>

>>>

>

>>> I've got the feeling this should work, but for some reason it isn't.

>

>>>

>

>>> So here is some code to test the problem with the errors describe in it:

>

>>>

>

>>>

>

>>>

>

>>> PRO testStrutRef

>

>>> Compile_Opt idl2

>

>>> a = list(findgen(10), /extract)

>

```

>>> a[4] = {value:3.0}
>
>>> help, a[4], /struct
>
>>> ;a[4].value = 4.0      ;won't work: % Attempt to store into an expression: >
>
>>> ;(a[4]).value = 4.0    ;won't work: % Attempt to store into an expression: Structure reference.
>
>>> ;((a)[4]).value = 4.0 ;won't work: % Attempt to store into an expression: Structure reference.
>
>>> struct = a[4]
>
>>> struct.value = 4.0
>
>>> a[4] = struct      ;works, but not really straight forward...
>
>>> print, a
>
>>> END
>
>>>
>
>>> It seems like I can't address a structure element of a list directly.
>
>>> Did I misplace my parenthesis? Is there a less cumbersome workaround?
>
>>
>
>> Seems like a flat-out bug to me. List elements should be "lvalues," as they say in the compile
business.
>
>> Craig
>
>>
>
>
>
> Yes, seems like a bug. Note that index notation works:
>
>
>
> IDL> print, (a[4]).(0)
>
>     3.00000
>
>
>
> Mike

```

>
> --
>
> Michael Galloy
>
> www.michaelgalloy.com
>
> Modern IDL: A Guide to IDL Programming (<http://modernidl.idldev.com>)
>
> Research Mathematician
>
> Tech-X Corporation

The index notation works only for printing, but not for assigning. But printing works also for print, `a[4].value`

By the way:

```
IDL> !version
{
  ARCH: "x86_64",
  OS: "Win32",
  OS_FAMILY: "Windows",
  OS_NAME: "Microsoft Windows",
  RELEASE: "8.3",
  BUILD_DATE: "Nov 15 2013",
  MEMORY_BITS: 64,
  FILE_OFFSET_BITS: 64
}
```

Cheers,
Helder
