Subject: structure references
Posted by spluque on Sun, 01 Dec 2013 18:33:03 GMT
View Forum Message <> Reply to Message

Hi,

While trying to set all elements of a structure to a value, I read
http://www.exelisvis.com/docs/Structure_References.html and it seems
this is not possible with a simple operation like:

x={a:double(0), b:double(0)}
x.(*)=20

Is writing a loop over the tags the only way?

Cheers,

--
Seb

Subject: Re: structure references
Posted by David Fanning on Sun, 01 Dec 2013 19:38:34 GMT
View Forum Message <> Reply to Message

Seb writes:

> While trying to set all elements of a structure to a value, I read
> http://www.exelisvis.com/docs/Structure_References.html and it seems
> this is not possible with a simple operation like:
>
> x={a:double(0), b:double(0)}
> x.(*)=20
>
> Is writing a loop over the tags the only way?

If all elements of your structure are the same data type, why aren't you
using an array?

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: structure references
Posted by spluque on Sun, 01 Dec 2013 19:46:35 GMT
View Forum Message <> Reply to Message

On Sun, 1 Dec 2013 12:38:34 -0700,
David Fanning <news@idlcoyote.com> wrote:

> Seb writes:
>> While trying to set all elements of a structure to a value, I read
>> http://www.exelisvis.com/docs/Structure_References.html and it seems
>> this is not possible with a simple operation like:

>> x={a:double(0), b:double(0)} x.(*)=20

>> Is writing a loop over the tags the only way?

> If all elements of your structure are the same data type, why aren't
> you using an array?

I'd like to have easier access to the elements of such array. It would
make life much simpler to be able to call a particular element by name
(e.g. tag) than by array index. Several routines need to access to
these data, so using an array index for each element soon makes code
hard to read.

Thanks,

--
Seb

---

Subject: Re: structure references
Posted by David Fanning on Sun, 01 Dec 2013 19:49:31 GMT
View Forum Message <> Reply to Message

Seb writes:

> I'd like to have easier access to the elements of such array. It would
> make life much simpler to be able to call a particular element by name
> (e.g. tag) than by array index. Several routines need to access to
> these data, so using an array index for each element soon makes code
> hard to read.

"Ah," he says, as he shakes his head.

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: structure references
Posted by wlandsman on Sun, 01 Dec 2013 19:52:25 GMT
View Forum Message <> Reply to Message

On Sunday, December 1, 2013 1:33:03 PM UTC-5, Sebastian Luque wrote:
> Hi,
>
In general, a structure contains elements of different types (e.g. string types) so

x.(*)=20

isn't allowed. Instead of a structure, you might look at the HASH() syntax which allows one to specify multiple hashes in a single call:

```
IDL> a = hash('a',double(0),'b',double(0), c:'string' )
IDL> a[['a','b']] = 20.0d
IDL> print,a
c: string field
a:       20.000000
b:       20.000000
```

>
> While trying to set all elements of a structure to a value, I read
>
> http://www.exelisvis.com/docs/Structure_References.html and it seems
>
> this is not possible with a simple operation like:
>
>
>
> x={a:double(0), b:double(0)}
>
> x.(*)=20
>

---

Subject: Re: structure references
Posted by spluque on Mon, 02 Dec 2013 01:18:36 GMT
View Forum Message <> Reply to Message

On Sun, 1 Dec 2013 11:52:25 -0800 (PST),
wlandsman <wlandsman@gmail.com> wrote:

> On Sunday, December 1, 2013 1:33:03 PM UTC-5, Sebastian Luque wrote:
>> Hi,

> In general, a structure contains elements of different types
> (e.g. string types) so

> x.(*)=20

> isn't allowed. Instead of a structure, you might look at the HASH()
> syntax which allows one to specify multiple hashes in a single call:

IDL> a = hash('a',double(0),'b',double(0), c:'string' ) a[['a','b']] =
IDL> 20.0d print,a
> c: string field a: 20.000000 b: 20.000000

Thanks, hashes do seem attractive for these purposes, however it's a
relatively new data type, so would require major changes in other parts
of the code. I might have to bite the bullet and stick to arrays, and
having code that looks like this:

```
FUNCTION FOO, A
   return, [A*2, A^2, A^0.5]
END

;; Option 1
PRO BAR, B
   x=foo(b)
   x_dbl=x[0]
   x_sqr=x[1]
   x_sqrt=x[2]
   ... LOTS OF OPERATIONS WITH x_dbl, x_sqr, x_sqrt, x[*]
END
;; Option 2
PRO BAR, B
   x=foo(b)
   ... LOTS OF OPERATIONS WITH x[0], x[1], x[2], x[*]
END
```

No problem, I personally prefer something like the latter but with a
little more meaningful nomenclature.

--

Seb

---

Subject: Re: structure references
Posted by Matthew Argall on Mon, 02 Dec 2013 04:44:14 GMT
View Forum Message <> Reply to Message

> FUNCTION FOO, A
>
> return, [A*2, A^2, A^0.5]
>
> END

Why not this?

```
PRO FOO, A, TwoA, ASquare, ASqrt
  TwoA = 2*A
  ASquare = A^2
  ASqrt = Sqrt(A)
END
```

---

Subject: Re: structure references
Posted by spluque on Thu, 05 Dec 2013 19:20:20 GMT
View Forum Message <> Reply to Message

On Sunday, December 1, 2013 10:44:14 PM UTC-6, Matthew Argall wrote:
>> FUNCTION FOO, A
>
>>
>
>> return, [A*2, A^2, A^0.5]
>
>>
>
>> END
>
>
>
>
>
> Why not this?
>
>
>
> PRO FOO, A, TwoA, ASquare, ASqrt

>
>    TwoA = 2*A
>
>    ASquare = A^2
>
>    ASqrt = Sqrt(A)
>
> END

Thanks, this is probably the best thing to do.  I have about 25 variables that would be output from FOO, so it would require a lengthy call.

Cheers,
Seb

---