
Subject: Re: Structures:

Posted by [Mark Hadfield](#) on Wed, 21 Feb 2001 20:52:42 GMT

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

"Chris Bull" <cjbull@another.com> wrote in message

news:owUk6.10166\$5n4.162849@news6-win.server.ntlworld.com...

- > A question regarding IDL structures, I have an application that I am
- > writing that I am using a very complex
- > structure as a Uvalue to hold my data in, as this application evolves I am
- > wondering whether it
- > is possible to add 'branches' to a structure after it is defined (and set
- as
- > a uvalue), from within a separate subroutine?

Well, yes and no. The IDL built-in CREATE_STRUCT allows you to add tags to a structure, but what it actually does is erase the original value and generate a new one, i.e. a full copy of the structure data in memory. This is fine for small structures, but inefficient on large ones. It is a consequence of the decisions IDL made when they designed the structure data type: keep it contiguous (except for padding) in memory, simple and fast.

- > I'm Thinking along the lines of describing an application that can hold many
- > data sets, of different types in memory
- > and load them up separately (much like a imaging program or multiple
- > documents in word) except the data sets are
- > very complex and of differing structures

You could use a data container based on pointers, but accessing and modifying this will get complicated. Me, I get confused by all those dereferencing symbols, e.g. `*(c(*a.b))`

So my recommendation is to use an object-based data container. There are no adequate ones supplied with IDL (more's the pity) so you will have to write your own or adapt someone else's. For a start you could look at a few on my WWW page: go to <http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/> and look at:

```
mgh_vector__define
mgh_queue__define
mgh_stack__define
```

The vector is the one that might be of use. It is a 1-D vector that can hold data elements of any type, indexed by number. The size is adjusted dynamically & automatically as elements are added.

A few months ago I tried writing a similar "dictionary" object, i.e. one that holds a series of name:value pairs (rather like an IDL structure.) I

found that accessing the data elements was too slow. Such data structures are common and provide reasonable performance in other languages (e.g. Python, Matlab) so it should be possible in principle to create one in IDL, but you've got to get clever in the way you look up the names.

--

Mark Hadfield
m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield>
National Institute for Water and Atmospheric Research

Subject: Re: Structures:
Posted by [Jason P. Meyers](#) on Wed, 21 Feb 2001 21:58:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Try using anonymous structures. Here is a quick and dirty demo that illustrates it can be done. Note, you can even store structures within anonymous structures!

I hope this helps.

Example:

```
pro test_struct, data
```

```
  If N_Elements(data) gt 0 then print, data
```

```
  data = {a:3, b:'Hello World', c:indgen(3,3)}
```

```
  print, "  
  help, data  
  print, 'Data is now:'  
  print, data
```

```
  data = {a:4, b:'Hello World', c:indgen(4,4), d:indgen(3,3)}
```

```
  print, "  
  help, data  
  print, 'Data is now:'  
  print, data
```

```
end
```

--

Jason Meyers
Ph.D. Student, Center for Imaging Science
Rochester Institute of Technology

Subject: Re: Structures:
Posted by [R.Bauer](#) on Thu, 22 Feb 2001 07:24:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Chris Bull wrote:

>
> Hi all,
>
> A question regarding IDL structures, I have an application that I am
> writing that I am using a very complex
> structure as a Uvalue to hold my data in, as this application evolves I am
> wondering whether it
> is possible to add 'branches' to a structure after it is defined (and set as
> a uvalue), from within a separate subroutine?
>
> I'm Thinking along the lines of describing an application that can hold many
> data sets, of different types in memory
> and load them up separately (much like a imaging program or multiple
> documents in word) except the data sets are
> very complex and of differing structures
>
> Any Ideas?

Yes it's possible if you have anonymous structure.

the trick is to store all structure levels to a string vector
and all tagvalues to a vector of pointers.
Then the new values in both are inserted and the structure is
new generated.

This is quite fast.

For example add_tag

```
; d = {A: 1, B: {B1: 0, B2: 1}, C: {B1: 0, B2: 1}}  
;  
;  
; help,d,/str  
; ** Structure <1331608>, 3 tags, length=10, refs=1  
; A      INT      1  
; B      STRUCT   -> <Anonymous> Array[1]  
; C      STRUCT   -> <Anonymous> Array[1]  
  
; result = add_tag(d, 'ASA', 11, sub = 'B')
```

```
; HELP, result.b, /STR
; ** Structure <133ce88>, 3 tags, length=6, refs=1
; B1      INT      0
; B2      INT      1
; ASA     INT      11

; result = add_tag(d, 'BSB', {C11: INDGEN(3)}, sub = 'B.B3')
; HELP, result.b, /STR
; ** Structure <1357758>, 3 tags, length=10, refs=2:
; B1      INT      0
; B2      INT      1
; B3      STRUCT   -> <Anonymous> Array[1]
```

Please have a look at the following routines.

```
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_
html/dbase/download/replace_tagvalue.tar.gz
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/add_tag.tar.gz
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/rename_tag.tar.gz
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/delete_tag.tar.gz
```

For further routines and licensing please look at
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.html

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
<http://www.fz-juelich.de/icg/icg1/>
=====

a IDL library at Forschungszentrum Jülich
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.html

<http://www.fz-juelich.de/zb/text/publikation/juel3786.html>

Subject: Re: Structures:
Posted by [Chris Bull](#) on Fri, 23 Feb 2001 19:37:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks all, some useful Ideas here to ponder on!

Cheers :*)

"Chris Bull" <cjbull@another.com> wrote in message
news:owUk6.10166\$5n4.162849@news6-win.server.ntlworld.com...

> Hi all,

>

> A question regarding IDL structures, I have an application that I am
> writing that I am using a very complex
> structure as a Uvalue to hold my data in, as this application evolves I am
> wondering whether it
> is possible to add 'branches' to a structure after it is defined (and set
as

> a uvalue), from within a separate subroutine?

>

> I'm Thinking along the lines of describing an application that can hold
many

> data sets, of different types in memory

> and load them up separately (much like a imaging program or multiple
> documents in word) except the data sets are

> very complex and of differing structures

>

> Any Ideas?

>

> Sorry I don't have the code here! as this is my home PC not work

>

> Regards

>

>

> Chris Bull

>

>

>

>

>