
Subject: Re: Removing fields from a structure
Posted by [Brian Larsen](#) on Thu, 01 Feb 2007 17:31:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

I don't know that it is more elegant but it is built into solarsoft as
rem_tag.pro

```
[larsen@titan]~> cat /usr/local/ssw/gen/idl/struct/rem_tag.pro
;+
; Project   : HESSI
;
; Name      : REM_TAG
;
; Purpose   : remove tags from a structure
;
; Syntax    : NEW_STRUCTURE=REM_TAG(STRUCT,TAG)
;
; Inputs    : STRUCT = input structure (array or scalar)
;           : TAG = array of tag names or indexes to remove
;
; Outputs   : NEW_STRUCTURE = new structure
;
; Keywords  : NAME = new name for structure (use with care)
;
; Category  : Structure handling
;
; Written   : 1-Dec-02, Zarro (EER/GSFC)
;
; Modified  : 24-Dec-04, Zarro (L-3Com/GSFC)
;           - vectorized, and removed dangerous /free_mem
;           01-Feb-05, Csillaghy (Univ. Applied Sciences NW
Switzerland)
;           - changed n_elements( struct ) to size(struct /dim ),
see at the end.
;-

function
rem_tag,struct,tags,_extra=extra,quiet=quiet,err=err,name=na me

verbose=1-keyword_set(quiet)

;-- bypass for older IDL versions

forward_function rem_tag2

if (1-since_version('5.4')) then begin
if have_proc('rem_tag2') then $
return,rem_tag2(struct,tags,_extra=extra,name=name,err=err)
```

```
message,'no longer supported for this version of IDL - '+'!  
version.release,/cont  
if exist(struct) then return,struct else return,-1  
endif
```

```
;- catch errors
```

```
err="  
error=0  
catch,error  
if error ne 0 then begin  
err=err_state()  
if verbose then message,err,/cont  
catch,/cancel  
if exist(struct) then return,struct else return,-1  
endif
```

```
if (1-is_struct(struct)) then begin  
err='invalid input'  
pr_syntax,'new_struct=rem_tag(struct,tag_name)'  
if exist(struct) then return,struct else return,-1  
endif
```

```
sz=size(tags,/type)  
index_input=(sz gt 1) and (sz lt 6)  
string_input=(sz eq 7)  
err='input tag names or tag indexes required'  
if (not index_input) and (not string_input) then begin  
if verbose then message,err,/cont & return,struct  
endif  
if string_input and is_blank(tags) then begin  
if verbose then message,err,/cont & return,struct  
endif
```

```
;- create structure template with tags removed
```

```
err="  
stag_names=tag_names(struct)  
ntags=n_elements(stag_names) & stag_index=lindgen(ntags)  
for i=0,ntags-1 do begin  
if string_input then chk=where(strup(tags) eq stag_names[i],count)  
else $  
chk=where(long(tags) eq stag_index[i],count)  
if count eq 0 then begin  
if is_struct(temp) then $  
temp=create_struct(temp,stag_names[i],struct[0].(i)) else $  
temp=create_struct(stag_names[i],struct[0].(i))  
endif
```

```
endifor

;-- all tags removed

if not is_struct(temp) then return,-1

;-- no tags removed

rtags=tag_names(temp)
if n_elements(rtags) eq ntags then return,struct

;-- rename if requested

if is_string(name) then temp=create_struct(temp,name=strup(name))

dims_arr =size(struct, /dim)
if total(dims_arr) GT 1 then $
temp=replicate2(temporary(temp),dims_arr)

struct_assign,struct,temp,/nozero

return,temp

end
```

Brian

Brian A. Larsen
Dept. of Physics
Space Science and Engineering Lab (SSEL)
Montana State University - Bozeman
Bozeman, MT 59717

On Feb 1, 10:27 am, "M. Katz" <MKatz...@yahoo.com> wrote:
> I was wondering if there's a simple way to remove a field from a
> structure?
>
> I can envision a manual way:
> 0) creating a new structure variable for the result
> 1) reading the original's tag names
> 2) stepping through tag by tag using indexed value references, like a.
> (i)
> 3) using create_struct() to put the tags and values into the new

- > structure, one by one, except for the tag(s) we're removing.
 - >
 - > Does anyone have a more elegant suggestion?
 - > M. Katz
-

Subject: Re: Removing fields from a structure

Posted by [Allan Whiteford](#) on Fri, 02 Feb 2007 02:48:31 GMT

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

M. Katz wrote:

- > I was wondering if there's a simple way to remove a field from a
- > structure?
- >
- > I can envision a manual way:
- > 0) creating a new structure variable for the result
- > 1) reading the original's tag names
- > 2) stepping through tag by tag using indexed value references, like a.
- > (i)
- > 3) using `create_struct()` to put the tags and values into the new
- > structure, one by one, except for the tag(s) we're removing.
- >
- > Does anyone have a more elegant suggestion?
- > M. Katz
- >

Sorry, nothing more elegant from me. But here's some code I wrote a while ago which I think pretty much does what you've suggested.

```
pro remove_tag,struct,tagname
```

```
  searchtag=strupcase(tagname)
  tagnames=tag_names(struct)
  a=[-1]
  if n_elements(tagname) eq 1 then a=[a,where(tagnames $
ne searchtag)] else for i=0,n_elements(tagnames)-1 do $
if (where(searchtag eq tagnames[i]))[0] eq -1 then a=[a,i]
  if n_elements(a) eq 1 then return
  if a[1] eq -1 then return
  newstruct=create_struct(tagnames[a[1]],struct.(a[1]))
  if n_elements(a) gt 2 then for i=2,n_elements(a)-1 $
do newstruct=create_struct(newstruct,tagnames[a[i]],struct.(a[i ]))
  struct=newstruct
end
```

tagname can be a string or string array. Removing multiple tags by calling the routine multiple times was a real time overhead which is why you can give it an array of tags. struct is the input structure which

will be modified on output.

Sorry about the lack of documentation, poor formatting and calling an important variable 'a'. I never distributed this code - it's still sitting around waiting to be put in to CVS to replace the old version which could only do one tag at a time.

The SSW code is probably better and I'm sure the loop gurus can remove the loops and possibly not call create_struct multiple times which is the big CPU time overhead.

Thanks,

Allan

Subject: Re: Removing fields from a structure
Posted by [R.Bauer](#) on Fri, 02 Feb 2007 10:30:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

M. Katz wrote:

- > I was wondering if there's a simple way to remove a field from a
- > structure?
- >
- > I can envision a manual way:
- > 0) creating a new structure variable for the result
- > 1) reading the original's tag names
- > 2) stepping through tag by tag using indexed value references, like a.
- > (i)
- > 3) using create_struct() to put the tags and values into the new
- > structure, one by one, except for the tag(s) we're removing.
- >
- > Does anyone have a more elegant suggestion?
- > M. Katz
- >

Hi

so here are some very fast ones based on pointer operations which do work on sub structures too

delete_tag

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl_html/dbase/delete_tag_dbase.pro.html

add_tag

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl_html/dbase/add_tag_dbase.pro.html

replace_tag_value
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl_html/dbase/replace_tagvalue_dbase.pro.html

rename_tag
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl_html/dbase/rename_tag_dbase.pro.html

If you like to read examples in german may be you are interested in reading a book about our library
http://www.fz-juelich.de/zb/datapool/page/439/00322_Bauer.pdf

cheers Reimar

--

Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

a IDL library at Forschungszentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro.html
=====

Subject: Re: Removing fields from a structure
Posted by [R.Bauer](#) on Fri, 02 Feb 2007 10:31:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
; EXAMPLE:
; d = {A: 1, B: {B1: 0, B2: 1}, C: {B1: 0, B2: 1}}
;
; result = delete_tag(d, 'B')
; help, result, /str
; ** Structure <1342678>, 2 tags, length=8, refs=1:
; A      INT      1
; C      STRUCT   -> <Anonymous> Array[1]
;
; result = delete_tag(d, ['B*'], SUB = 'C')
; help, result, /str
; ** Structure <13694a8>, 2 tags, length=6, refs=1:
; A      INT      1
; B      STRUCT   -> <Anonymous> Array[1]
;
; result = delete_tag(d, '*B1', SUB = ['B', 'C'])
; help, result.b, /str
; ** Structure <1331be8>, 1 tags, length=2, refs=1:
```

```

; B2      INT      1
;
; result = delete_tag(d, ['B1', 'B2'], /ALL)
; help, result, /str
; ** Structure <170fac8>, 1 tags, length=2, refs=1:
; A      INT      1
;
; help,(delete_tag(d,'b1',sub_structure='B')).b,/str
; B2      INT      1
;
; help,delete_tag(d,'B',/all)
; ** Structure <140aa40>, 2 tags, length=6, refs=1:
; A      INT      1
; C      STRUCT   -> <Anonymous> Array[1]
;
; help,(delete_tag(d,'b1',sub_structure='B')).c,/str
; ** Structure <1333618>, 2 tags, length=4, refs=1:
; B1      INT      0
; B2      INT      1
;
; help,(delete_tag(d,'*',sub_structure='B')),/str
; ** Structure <133b2d8>, 2 tags, length=6, refs=1:
; A      INT      1
; C      STRUCT   -> <Anonymous> Array[1]
;
; help,(delete_tag(d,'*',sub_structure='B',/sort)),/str
; ** Structure <d40daa8>, 2 tags, length=6, refs=1:
; A      INT      1
; C      STRUCT   -> <Anonymous> Array[1]
;
; help,(delete_tag(d,sub_structure=['B','A'])),/str
; ** Structure <132fe78>, 2 tags, length=6, refs=1:
; A      INT      1
; C      STRUCT   -> <Anonymous> Array[1]
;
; help,(delete_tag(d,sub_structure=['B','C'])),/str
; ** Structure <1333288>, 1 tags, length=2, refs=1:
; A      INT      1
;
; help,(delete_tag(d,['B1','B2'],sub_structure=['B','C'])),/str
; ** Structure <46a9498>, 1 tags, length=2, refs=1:
; A      INT      1
;
; d = {ABRA: {B1: 0, B2: 1}, ABRA2B: {B1: 0, B2: 1}, ABRA3B: {B1: 0,
B2: 1}}
;
; result=delete_tag(d,'abra')
; help,result,/str

```

```

; ** Structure <1365a48>, 2 tags, length=8, refs=1:
; ABRA2B      STRUCT  -> <Anonymous> Array[1]
; ABRA3B      STRUCT  -> <Anonymous> Array[1]
;
; result=delete_tag(d,'abra*')
; help,result,/str
; RESULT      INT      =      -1
;
; result=delete_tag(d,'abra?B')
; help,result,/str
;
; ** Structure <1364358>, 1 tags, length=4, refs=1:
; ABRA        STRUCT  -> <Anonymous> Array[1]
;
; d = {ABRA: {B1: 0, B2: {A:1}}, ABRA2B: {B1: 0, B2: {A:1}}, ABRA3B:
{B1: 0, B2: {A:1}}}
; result=delete_tag(d,'A',sub_structure='abra.b2')
; help,result.abra,/str
; ** Structure <1365c68>, 1 tags, length=2, refs=2:
; B1          INT      0
;

```

Reimar Bauer wrote:

> M. Katz wrote:

>

>> I was wondering if there's a simple way to remove a field from a
>> structure?

>>

>> I can envision a manual way:

>> 0) creating a new structure variable for the result

>> 1) reading the original's tag names

>> 2) stepping through tag by tag using indexed value references, like a.

>> (i)

>> 3) using create_struct() to put the tags and values into the new

>> structure, one by one, except for the tag(s) we're removing.

>>

>> Does anyone have a more elegant suggestion?

>> M. Katz

>>

>

>

> Hi

>

>

> so here are some very fast ones based on pointer operations which do

> work on sub structures too

>

> delete_tag

> http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl_html/dbase/delete_tag_dbase.pro.html
>
> add_tag
> http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl_html/dbase/add_tag_dbase.pro.html
>
> replace_tag_value
> http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl_html/dbase/replace_tagvalue_dbase.pro.html
>
> rename_tag
> http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl_html/dbase/rename_tag_dbase.pro.html
>
>
> If you like to read examples in german may be you are interested in
> reading a book about our library
> http://www.fz-juelich.de/zb/datapool/page/439/00322_Bauer.pdf
>
> cheers Reimar
>

--

Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro.html

=====