Subject: Re: Renaming tags in an array of structures
Posted by David Fanning on Thu, 16 Sep 2010 18:50:59 GMT
View Forum Message <> Reply to Message

Michael Williams writes:

> Can anyone see how to fix these problems so that the function works
> for arbitrarily many tag names, copies the data (ideally without using
> execute statements, which is the only way I can think of doing it),
> and the old fields are deleted? Or is there another solution
> altogether?

Yes, I think I can see how to do it. I can't see how to
do it in a couple of lines of code. And I can't see how
to do it without either using EXECUTE or FOR loops. But
I think it can be done.

Is this worth something to you? :-)

This would be a couple of hours worth of effort to
write a completely general program. For example,
you will probably have to write some kind of
recursive function to copy tag values that are
pointers and/or structures, etc. Just that alone
will take a couple hours, probably, although I
used to have something like that laying around
here.

Cheers,

David

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

Subject: Re: Renaming tags in an array of structures
Posted by penteado on Thu, 16 Sep 2010 19:05:30 GMT
View Forum Message <> Reply to Message

On Sep 16, 3:28 pm, Michael Williams <mjwilli...@gmail.com> wrote:
> I have an array of structures:
>
> IDL> x = {a: 1, b:2, c: indgen(5)}
> IDL> xx = replicate(x, 10)

>
> In this case, I want to rename the 'b' tag to 'foo' and the 'c' tag to
> bar. But I want a function that works for arbitrary tag names. I have
> readhttp://www.dfanning.com/code_tips/addfield.html, which describes
> how to add a field to an array of structures using struct_assign. So
> far I have the following function, which seems like it's headed in the
> right direction: it will add the foo and bar tags. But (i) it only
> works if you're renaming exactly two tags because of the create_struct
> call (ii) it doesn't copy the data in the old tags to the new tags and
> (iii) it doesn't delete the old tags, which is necessary for true
> "replacement".
>
> Can anyone see how to fix these problems so that the function works
> for arbitrarily many tag names, copies the data (ideally without using
> execute statements, which is the only way I can think of doing it),
> and the old fields are deleted? Or is there another solution
> altogether?

Any reason not to do it with hashes and avoid all those troubles?

Subject: Re: Renaming tags in an array of structures
Posted by Michael Williams on Thu, 16 Sep 2010 20:33:28 GMT
View Forum Message <> Reply to Message

On Sep 16, 9:05 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> Any reason not to do it with hashes and avoid all those troubles?

I didn't actually know those had arrived with 8.0. I'm still on 7.x. I would probably migrate the code to Python rather than do upgrade and rewrite. Thanks for the heads-up though.

Subject: Re: Renaming tags in an array of structures
Posted by Michael Williams on Thu, 16 Sep 2010 20:40:42 GMT
View Forum Message <> Reply to Message

On Sep 16, 8:50 pm, David Fanning <n...@dfanning.com> wrote:
>> Can anyone see how to fix these problems so that the function works
>> for arbitrarily many tag names, copies the data (ideally without using
>> execute statements, which is the only way I can think of doing it),
>> and the old fields are deleted? Or is there another solution
>> altogether?
>
> Yes, I think I can see how to do it. I can't see how to
> do it in a couple of lines of code. And I can't see how
> to do it without either using EXECUTE or FOR loops. But

> I think it can be done.

Thanks. If there isn't a more elegant way (short of migrating to hashes or a post-Cold War language ;-) then I think I can see a way of doing it too. Luckily, I don't need to worry about nested structures. I'm just dealing with what is essentially a 2-D matrix where the columns have names and each element is a float, so I can afford to be fairly braindead about this.

> Is this worth something to you? :-)

I'm afraid I'm a PhD student with (i) no money (ii) no time! :-) I'll post my solution, and comments (and improvements) are of course appreciated!

-- Mike

Subject: Re: Renaming tags in an array of structures
Posted by Michael Williams on Thu, 16 Sep 2010 21:13:20 GMT
View Forum Message <> Reply to Message

On Sep 16, 10:40 pm, Michael Williams <mjwilli...@gmail.com> wrote:
> On Sep 16, 8:50 pm, David Fanning <n...@dfanning.com> wrote:
>
>>> Can anyone see how to fix these problems so that the function works
>>> for arbitrarily many tag names, copies the data (ideally without using
>>> execute statements, which is the only way I can think of doing it),
>>> and the old fields are deleted? Or is there another solution
>>> altogether?
>
>> Yes, I think I can see how to do it. I can't see how to
>> do it in a couple of lines of code. And I can't see how
>> to do it without either using EXECUTE or FOR loops. But
>> I think it can be done.
>
> Thanks. If there isn't a more elegant way (short of migrating to
> hashes or a post-Cold War language ;-) then I think I can see a way of
> doing it too. Luckily, I don't need to worry about nested structures.
> I'm just dealing with what is essentially a 2-D matrix where the
> columns have names and each element is a float, so I can afford to be
> fairly braindead about this.

And here is my implementation. I am obviously in a state of sin with all these for loops and execute statements, but it works for my purposes. Comments welcome.

The one practical limitation that may cause me some problems in the

future is that fact that every element of the new array of structures is forced to be a double because of the way I'm deleting the old_tags. I can see a rather tedious and ugly way around this (essentially constructing a lookup table which says what string to add to the empties variable based on the type of the each tag you keep). That solution seems ugly even for IDL so I would be particularly interested in any ideas there.

```
function rename_tags_array, str, old_tags, new_tags
   nchange = n_elements(old_tags)

   ; Set up new template with tags in str + new_tags
   empties = '0d'
   for i = 1, nchange - 1 do empties += ',0d'
   void = execute('template = create_struct(new_tags, ' + empties +
', str[0])')
   newstr = replicate(template, n_elements(str))
   struct_assign, str, newstr

   ; Copy str.old_tags to newstr.new_tags
   for i = 0, nchange - 1 do begin
      void = execute('newstr.' + new_tags[i] + ' = str.' +
old_tags[i])
   endfor

   ; Delete newstr.old_tags
   ;
   ; (i) generate list of all tags in newstr
   tags = tag_names(newstr)
   ; (ii) remove old_tags from this list
   for i = 0, nchange - 1 do begin
      j = where(strupcase(tags) eq strupcase(old_tags[i]),
complement = keep)
      tags = tags[keep]
   endfor
   ; (iii) setup template structure
   empties = '0d'
   for i = 1, n_elements(tags) - 1 do empties += ',0d'
   void = execute('template = create_struct(tags, ' + empties + ')')
   stop
   newstr2 =  replicate(template, n_elements(str))
   ; (iv) Throw out old old_tags
   struct_assign, newstr, newstr2

   return, newstr2
end
```

Subject: Re: Renaming tags in an array of structures
Posted by Michael Williams on Thu, 16 Sep 2010 21:18:52 GMT
View Forum Message <> Reply to Message

On Sep 16, 11:13 pm, Michael Williams <mjwilli...@gmail.com> wrote:
[...]
> ; (iii) setup template structure
> empties = '0d'
> for i = 1, n_elements(tags) - 1 do empties += ',0d'
> void = execute('template = create_struct(tags, ' + empties + ')')
> stop
> newstr2 = replicate(template, n_elements(str))
> ; (iv) Throw out old old_tags
> struct_assign, newstr, newstr2
>
> return, newstr2
> end

pro-tip for those trying to run this code: you're going to want to
delete that "stop" ;-)

Subject: Re: Renaming tags in an array of structures
Posted by Paul Van Delst[1] on Fri, 17 Sep 2010 13:30:05 GMT
View Forum Message <> Reply to Message

Michael Williams wrote:
> I have an array of structures:
>
> IDL> x = {a: 1, b:2, c: indgen(5)}
> IDL> xx = replicate(x, 10)
>
> In this case, I want to rename the 'b' tag to 'foo' and the 'c' tag to
> bar.

Can one inquire why you would like/need to do this?

Subject: Re: Renaming tags in an array of structures
Posted by Michael Williams on Fri, 17 Sep 2010 14:16:03 GMT
View Forum Message <> Reply to Message

On Sep 17, 3:30 pm, Paul van Delst <paul.vande...@noaa.gov> wrote:
> Michael Williams wrote:
>> I have an array of structures:
>
>> IDL> x = {a: 1, b:2, c: indgen(5)}
>> IDL> xx = replicate(x, 10)

>
>> In this case, I want to rename the 'b' tag to 'foo' and the 'c' tag to
>> bar.
>
> Can one inquire why you would like/need to do this?

I'm reading in a FITS file (data format used in astronomy) that I did not create. The library function I'm using to do this (mrdfits from astroidl) returns an array of structures. The names of the tags are specified in the fits file itself.

It so happens that in this new file I'm working with, these names do not match the choices used for the same data in all my other files. For example, let's say I want to plot old_data.speed against new_data.speed. But new_data doesn't have a .speed tag. It calls it .velocity. If it's just one tag and just one case, then hard coding something would work, but as the number of tags grows this becomes ugly and error prone. It would be much easier and more general to just rename tags. Unfortunately that seems to be non-trivial in IDL.

In my case, the best thing to do is probably to edit the FITS file at source and change the header so that the tag names match. That may not be possible (or desirable from a data purity point of view) in all circumstances, and renaming data structures or changing the metadata associated with a strucure should of course be possible in a modern language! Hence this question :-)

-- Mike

Subject: Re: Renaming tags in an array of structures
Posted by wlandsman on Fri, 17 Sep 2010 14:50:35 GMT
View Forum Message <> Reply to Message

Check out the ALIAS keyword in mrdfits.pro

```
;       ALIAS    The keyword allows the user to specify the column
names
;                to be created when reading FITS data.  The value of
;                this keyword should be a 2xn string array.  The first
;                value of each pair of strings should be the desired
;                tag name for the IDL column.    The second should be
;                the FITS TTYPE value.
```

This is probably more efficient than changing the tag names after the structure has been read, though I realize that you might not know the names you need until after the file is read. (And I haven't used the ALIAS keyword much so I can't vouch for it.) --Wayne

On Sep 17, 10:16 am, Michael Williams <mjwilli...@gmail.com> wrote:
>
> I'm reading in a FITS file (data format used in astronomy) that I did
> not create. The library function I'm using to do this (mrdfits from
> astroidl) returns an array of structures. The names of the tags are
> specified in the fits file itself.
>
> It so happens that in this new file I'm working with, these names do
> not match the choices used for the same data in all my other files.
> For example, let's say I want to plot old_data.speed against
> new_data.speed. But new_data doesn't have a .speed tag. It calls
> it .velocity. If it's just one tag and just one case, then hard coding
> something would work, but as the number of tags grows this becomes
> ugly and error prone. It would be much easier and more general to just
> rename tags. Unfortunately that seems to be non-trivial in IDL.
>
> In my case, the best thing to do is probably to edit the FITS file at
> source and change the header so that the tag names match. That may not
> be possible (or desirable from a data purity point of view) in all
> circumstances, and renaming data structures or changing the metadata
> associated with a strucure should of course be possible in a modern
> language! Hence this question :-)
>
> -- Mike
