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
```