

---

Subject: Combing structures

Posted by [Michael Williams](#) on Mon, 07 Jun 2010 15:01:42 GMT

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

---

Hi,

I have two structures. Each contains a unique set of tags whose values are floats or arrays of floats. E.g.

```
xx = findgen(5)
y = 1.0
str1 = {one: xx, two:y}
```

```
xx = findgen(3)
y = -1.0
str2 = {three: xx, four: y}
```

I want to merge them into one flat structure of their elements. At the moment I am doing this manually:

```
str = {one:str1.one, two:str1.two, three:str2.three, four:str2.four}
```

This obviously becomes unwieldy and error-prone when my structures have several dozen tags. Is there a way of automating the merge step? I am happy to assume the tag names are unique (i.e. there is no danger of a collision).

I have played around with the tag\_names function, but I can't see a way of using this without using the EVALUATE function, which is generally a bad idea.

-- Mike

---

---

Subject: Re: Combing structures

Posted by [Michael Williams](#) on Mon, 16 Aug 2010 19:00:38 GMT

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

---

On 2010-06-07 17:01:42 +0200, Michael Williams said:

```
> xx = findgen(5)
> y = 1.0
> str1 = {one: xx, two:y}
>
> xx = findgen(3)
> y = -1.0
> str2 = {three: xx, four: y}
>
> I want to merge them into one flat structure of their elements. At the
```

> moment I am doing this manually:

I now have a related question. If

```
IDL> x = {a: [1,2], b: [3,4]}
```

```
IDL> y = {a: [5,6], b: [7:8]}
```

then doing

```
IDL> z = [x, y]
```

works, but yields, for example,

```
IDL> print, z.a
```

```
1    2
5    6
```

```
IDL> help, z.a
```

```
<Expression>  INT      = Array[2, 2]
```

I would like to concatenate two structures with matching tags by simply appending arrays, so that I get a structure in which

```
IDL> print, z.a
```

```
1    2    5    6
```

```
IDL> help, z.a
```

```
<Expression>  INT      = Array[4]
```

Is there a built-in way of doing this? The structures I am concatenating will always be defined such that the individual tags can be concatenated trivially like

```
IDL> a = [x.a, y.a]
```

```
IDL> print, a
```

```
1    2    5    6
```

but the only way I can think of doing this for the entire structure is by iterating over the tags and using the execute procedure [implementation after the signature], which is obviously spectacularly inelegant. Is there a better way?

-- Mike

[\*] Like this. test\_append\_struct shows it in use.

```
function append_struct, tx, ty
```

```
  tags = tag_names(tx)
```

```
  s = 'tmp = create_struct(' + tags[0] + ', [tx.' + tags[0] + '$  
    ', ty.' + tags[0] + '])'
```

```

void = execute(s)
for i = 1, n_elements(tags) - 1 do begin
    s = 'tmp = create_struct(tmp, "" + tags[i] + "", [tx.' + tags[i] + $
        ', ty.' + tags[i] + '])'
    void = execute(s)
endfor
return, tmp
end

```

```

pro test_append_struct
    p1 = {x: 0, y:1, z: 1.5}
    p2 = {x: -1, y:2, z: 4}
    p = append_struct(p1, p2)
    help, p, /struct
    print, p.x
    print, p.y
    print, p.z
    p = append_struct(p, p2)
    help, p, /struct
    print, p.x
    print, p.y
    print, p.z
end

```

---

Subject: Re: Combing structures

Posted by [penteado](#) on Mon, 16 Aug 2010 19:50:40 GMT

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

---

On Aug 16, 4:00 pm, Michael Williams <mjwilli...@gmail.com> wrote:

```

> function append_struct, tx, ty
>     tags = tag_names(tx)
>     s = 'tmp = create_struct("" + tags[0] + "", [tx.' + tags[0] + $
>         ', ty.' + tags[0] + '])'
>     void = execute(s)
>     for i = 1, n_elements(tags) - 1 do begin
>         s = 'tmp = create_struct(tmp, "" + tags[i] + "", [tx.' + tags[i] + $
>             ', ty.' + tags[i] + '])'
>         void = execute(s)
>     endfor
>     return, tmp
> end

```

That does not need to be so awkward. It could be done without execute, with just

```

function append_struct, tx, ty
    tags = tag_names(tx)

```

```
tmp=create_struct(tags[0],[tx.(0),ty.(0)])
for i = 1, n_elements(tags) - 1 do tmp=create_struct(tmp,tags[i],
[tx.(i),ty.(i)])
return, tmp
end
```

This could be made in a slightly easier way in IDL 8. But I do not see a way to concatenate in one line as you want.

---

---

Subject: Re: Combing structures

Posted by [Michael Williams](#) on Mon, 16 Aug 2010 19:55:35 GMT

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

---

On 2010-08-16 21:50:40 +0200, Paulo Penteado said:

```
> function append_struct, tx, ty
>   tags = tag_names(tx)
>   tmp=create_struct(tags[0],[tx.(0),ty.(0)])
>   for i = 1, n_elements(tags) - 1 do tmp=create_struct(tmp,tags[i],
> [tx.(i),ty.(i)])
>   return, tmp
> end
```

Thanks Paolo, that's certainly an improvement. I didn't know about the tx.(0) syntax (hence the executes).

-- Mike

---