Subject: IDL 6.1 create_struct() peculiarity Posted by Justin Bronn on Tue, 31 May 2005 20:00:50 GMT View Forum Message <> Reply to Message

So I'm working with an existing piece of software that uses the execute() function to resolve structures:

```
IDL> sname = 'xosimple'
IDL> stmt = 'struct = {' + sname + '}'
IDL> struct = execute(stmt)
```

I want to get rid of the execute() call because it is not allowed when running the VM (a nasty little error message will greet the VM user). I thought I was smart, and I found a way around this, by using the create_struct() function instead:

```
IDL> struct = create_struct(NAME=sname)
```

Now while this seems to work great at the IDL command line, when I try to embed this method into a callable procedure, the create_struct() throws a strange warning message (but doesn't stop the execution of the code):

% CREATE_STRUCT: Unexpected keyword cleanup stack found on return.

When trying this method out on IDL 5.6, I get the following error message:

% CREATE_STRUCT: Incorrect number of arguments.

So here's my stab at what I think is going on: RSI decided to allow the create_struct() method to be called with only the NAME keyword set in IDL 6.0, however they forgot to do all the testing so it doesn't clean up the C keyword arguments properly when called in a non-interactive mode.

Below is the code for a procedure that embeds this create_struct() method, and the sample structure definition file that should re-create this issue (FYI - I'm using IDL 6.1.1 on Linux - Fedora Core1).

Should I worry about this warning message? If anyone has any insight, please enlighten me & the newsgroup:)

```
Thanks,
-Justin
```

;; init_structs.pro

```
;; Routine that will initialize structures that have
;; their definition in a certain filename. Needs the
;; definition files to be in the same directory.
pro init_structs
 files = file_search('./xo*__define.pro', COUNT=count)
 if count eq 0 then $
  message, 'Could not find any structure definition files!'
 for i = 0L, count-1 do begin
   strings = stregex(files[i], '(xo.*)__define.pro' $
               , /EXTRACT, /SUBEXPR)
   struct_name = strings[1]
   struct = create_struct(NAME=struct_name)
 endfor
end
;; xosimple__define.pro
;; Simple structure definition file
pro xosimple__define
 temp = {xosimple, $
      title:", $
      int_value:0, $
      float value:0.0$
```

Subject: Re: IDL 6.1 create_struct() peculiarity
Posted by R.G. Stockwell on Wed, 01 Jun 2005 15:12:46 GMT
View Forum Message <> Reply to Message

```
"Justin Bronn" <jbronn@swri.edu> wrote in message
news:1117569650.819400.258050@g43g2000cwa.googlegroups.com...

> IDL> struct = create_struct(NAME=sname)

> Now while this seems to work great at the IDL command line, when I try

> to embed this method into a callable procedure, the create_struct()

> throws a strange warning message (but doesn't stop the execution of the

> code):
```

end

1

> % CREATE_STRUCT: Unexpected keyword cleanup stack found on return.

I have tried this, and I also get that warning message. The very odd thing though, is that I do NOT get the warning if I send in the string differently. i.e., the following works just fine.

namein = 'xosimple'
struct = create_struct(NAME=namein)

Perhaps it is something strange with the string being created with the stregex() function. Something internal to IDL, cause I cannot see the difference. In fact,

print, namein eq struct_name

returns a 1, indicating that the strings are identical. Very weird. I suggest sending in a bug report to RSI.

Cheers, bob

PS

DL> help,!version,/st

** Structure !VERSION, 8 tags, length=76, data length=76:

ARCH STRING 'x86'

OS STRING 'Win32'

OS_FAMILY STRING 'Windows'

OS NAME STRING 'Microsoft Windows'

RELEASE STRING '6.1.1'

BUILD_DATE STRING 'Oct 11 2004'

MEMORY_BITS INT 32

FILE OFFSET BITS

Page 4 of 4 ---- Generated from comp.lang.idl-pvwave archive