
Subject: Re: Dynamically naming structures in IDL
Posted by [Helder Marchetto](#) on Thu, 30 Jul 2015 14:32:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thursday, July 30, 2015 at 4:05:00 PM UTC+2, Scheherazade wrote:
> On Wednesday, July 29, 2015 at 5:16:43 PM UTC-4, wlandsman wrote:
>> Oops that should be
>>
>> status = execute("'save, ' + newname +', file =
>> '/Users/me/Documents/omni_general_'+year+'.sav' ")
>>>
>>
>> On Wednesday, July 29, 2015 at 4:46:02 PM UTC-4, wlandsman wrote:
>>> What if you also put the SAVE statement in execute()
>>>
>>> status = execute("save, newname, file = '/Users/me/Documents/omni_general_'+year+'.sav'
>>> ")
>>>
>>> On Wednesday, July 29, 2015 at 3:12:17 PM UTC-4, Scheherazade wrote:
>>>> On Wednesday, July 29, 2015 at 2:23:10 PM UTC-4, wlandsman wrote:
>>>> > If I understand correctly, you could use EXECUTE() to rename your structure:
>>>> >
>>>> > newname = 'new_structure_' + year
>>>> > status = execute(newname + '= TEMPORARY(new_variables)')
>>>> > save, newname, filename = '/Users/me/Documents/omni_general_'+year+'.sav'
>>>> >
>>>> >
>>>> > On Wednesday, July 29, 2015 at 1:08:52 PM UTC-4, Scheherazade wrote:
>>>> > > I have a structure, 'omni_gen', which is filled with data for 2011, 2012, and 2013. In my
program, I pull data out based on year (which is input) using a where statement, create variables
for the selected data, and run these variables through a routine which outputs a new set of
variables. I want to save these new variables to a structure, so I can later concatenate them
together to plot.
>>>> > >
>>>> > > I first save my structure to a different filename each time:
>>>> > >
>>>> > > year=strcompress(string(year), /remove_all)
>>>> > >
>>>> > > new_variables=create_struct('new_globtec_'+year, new_globtec, 'new_day_'+year,
new_day_numb, 'new_f10_'+year, new_f10_data, \$
>>>> > > 'new_ap_'+year, new_ap_data, 'new_smf10_'+year, new_smf10)
>>>> > >
>>>> > > save, new_variables, filename='/Users/me/Documents/omni_general_'+year+'.sav'
>>>> > >
>>>> > > When I go to concatenate the structures, I restore these files and manually rename the
corresponding new_variables structure each time, like so:
>>>> > >
>>>> > > restore, '/Users/keleuterio/Documents/omni_2011.sav'

```
>>>> > > new_variables=omni_data_2011
>>>> > >
>>>> > > But this obviously isn't automated. Is there a different way to do this so that the
structure itself will have a different name each time (ex: new_structure_2011,
new_structure_2012)?
>>>>
>>>> That successfully renamed the structure while in the program, but it doesn't work when I
restore the file later (since it saves newname, not the new structure).
>
> IDL responded with a "syntax error" on the save. I've been removing/adding quotation marks
within that line to fix it, but it gives the same error message each time. I also tried saving the two
commands as strings, then executing them (which is how execute is used in the Exelis Vis help
page http://www.exelisvis.com/docs/EXECUTE.html):
>
> status = execute( newname + '= TEMPORARY(new_variables)' )
> status1=('save' + "newname + file='/Users/keleuterio/Documents/omni_general_' + year +
'sav'")
> status2=execute("status1")
>
> But IDL responds with "% Attempt to call undefined procedure/function: 'STATUS1'". I think that
it's mostly a matter of putting the quotation marks in the right place, but I don't fully understand
how the Execute command recognizes the difference between a procedure (such as save) and a
string (such as the filepath).
```

Hi,
just my two cents. As far as I understand, the execute() function executes a string within the
parenthesis just as you would from the command line. So if you put:

```
status2=execute("status1")
```

you will get the same as if you did this:

```
IDL> status1
```

This obviously returns an error (not true in all cases, but for sure in your case). It is also very
handy to test things. Put the content of the execute statement in the command line and you can
test its functionality.

What you might want to be doing is:

```
status2=execute(status1)
```

That is without parenthesis. This way IDL puts within the parenthesis the save command.

Cheers,
Helder
