Subject: Re: Is there a work around to save multiple variables with same head but different tails without listing all of them?
Posted by Brian Daniel on Thu, 29 Sep 2011 12:45:20 GMT
View Forum Message <> Reply to Message

On Sep 29, 8:03 am, Dave Poreh <d.po...@gmail.com> wrote:
> On Sep 29, 4:50 am, "Brian J. Daniel" <Daniels...@yahoo.com> wrote:
>
>
>
>
>
>
>
>
>
>> On Sep 29, 5:10 am, voidspace <jhkim...@gmail.com> wrote:
>
>>> Hello folks,
>
>>> I have a question at beginner's level. When I saved multiple variables
>>> with same head but different tails in IDL, I found it is tedious to
>>> type all of them in as follows.
>
>>> SAVE, sst_cccma_cgcm3_1, sst_cnrm_cm3, sst_csiro_mk3_0,
>>> sst_gfdl_cm2_0, sst_gfdl_cm2_1, $
>>>   sst_iap_fgoals1_0_g, sst_inmcm3_0, sst_ipsl_cm4,
>>> sst_miroc3_2_medres, sst_mpi_echam5, $
>>>   sst_mri_cgcm2_3_2a, sst_ncar_ccsm3_0, sst_ncar_pcm1,
>>> sst_ukmo_hadcm3, sst_ukmo_hadgem1, $
>>>   FILENAME='sst_cmip3_20c3m.sav'
>
>>> The '/All' keyword may help, but there are more variables that I don't
>>> want to save in. This aroused my curiosity, so I post my question here
>>> to know whether there is a workaround.
>
>>> In fact, I tried to find an advanced way by myself, but found it's
>>> difficult to know without an expert's guidance.
>
>>> An attempt with my best knowledge is as follows. As all model names
>>> from 'cccma_cgcm3_1' to 'ukmo_hadgem1' are saved in a string array
>>> 'model', I considered the 'EXECUTE' command.
>
>>> One prior step I did was saving 15 variable names in one single string
>>> array 'sst'.
>
>>> IDL> sst=STRARR(N_elements(model))
>>> IDL> FOR i=0, N_elements(model)-1 DO sst[i]='sst_'+model[i]

>
>>> Now, 'sst' contains 15 different variable names that I want to save
>>> in. Then, tried the following
>
>>> IDL> result=EXECUTE("SAVE, FILENAME='sst_cmip3_20c3m.sav', "+sst[i])
>
>>> but in stuck because I have no idea how to make an implicit loop for
>>> 'sst[i]'.
>
>>> My attempt seems not a right way. I also thought it would be
>>> convenient if there is implicit do-loop like (sst(i),i=1,15) in
>>> Fortran, but I immediately got that it's nothing but a stupid idea.
>
>>> I thought over and googled as well but all were in vain.
>
>>> Can anybody suggest me a nice way to simplify the tedious command
>>> aforementioned?
>
>>> Best,
>>> John
>
>> You've already done the hard part, which is the string manipulation to
>> create your 'sst' array. The final step is to use StrJoin to bring
>> the array into one string joined by ', '. See below (UNTESTED).
>
>> result=EXECUTE("SAVE, FILENAME='sst_cmip3_20c3m.sav', strJoin(sst,',
>> ',/Single))
>
> Can you prepare a simple example? method seems quite good for me!
> Cheers,
> Dave

```
pro save_variables_example

;create simple variables
sst_cccma_cgcm3_1 = 1
sst_cnrm_cm3 = 2
sst_csiro_mk3_0 = 3
sst_gfdl_cm2_0 = 4
sst_gfdl_cm2_1 = 5
sst_iap_fgoals1_0_g = 6
sst_inmcm3_0 = 7
sst_ipsl_cm4 = 8
sst_miroc3_2_medres = 9
sst_mpi_echam5 = 10
sst_mri_cgcm2_3_2a = 11
sst_ncar_ccsm3_0 = 12
sst_ncar_pcm1 = 13
```

```
sst_ukmo_hadcm3 = 14
sst_ukmo_hadgem1 = 15

sst = ['sst_cccma_cgcm3_1', 'sst_cnrm_cm3', 'sst_csiro_mk3_0',$
 'sst_gfdl_cm2_0', 'sst_gfdl_cm2_1', 'sst_iap_fgoals1_0_g', $
 'sst_inmcm3_0', 'sst_ipsl_cm4', 'sst_miroc3_2_medres', $
 'sst_mpi_echam5', 'sst_mri_cgcm2_3_2a', 'sst_ncar_ccsm3_0', $
 'sst_ncar_pcm1', 'sst_ukmo_hadcm3', 'sst_ukmo_hadgem1']

Result = Execute('SAVE, FILENAME=''sst_cmip3_20c3m.sav'', '+$
 strJoin(sst,', ',/Single)+' ,/Verbose')

END
```