Subject: Re: write_png help
Posted by dg86 on Thu, 01 Dec 2016 11:24:20 GMT
View Forum Message <> Reply to Message

On Thursday, December 1, 2016 at 3:56:48 AM UTC-5, sid wrote:
> Hi all,
>   I have given below my program to create a series of frame*.png. But after doing this, s
number of frames are created. But all the frames are showing same data. I think there is some
problem with the code below, Can anyone please let me know what is the problem here.
>
>
> restore,'negdatanew.sav',/v
> s=size(ksom.data,/dim)
> for i=0,s(2)-1 do begin
> write_png,'frame'+string(i)+'.png',ksom(i).data
> write_png,'frame'+string(i)+'.png',tvrd(/true)
> endfor
>
>
> thanks

If you're doing this at the command line or in a batch file, the FOR loop has to be
written as a single line of code.
Each of the commands within the FOR loop, furthermore, has to end with
the "command termination" character, '&'. A loop that loops over three
commands could be written on a single line as

for i = 0, s[2]-1 do begin command1 & command2 & command3 & endfor

You can avoid having a long ugly line of code by using the "line continuation" character, '$'.
In that case, my example could be formatted as

for i = 0, s[2]-1 do begin $
  command1 & $
  command2 & $
  command3 & $
endfor

Notice that there's only a continuation character after BEGIN. You don't want to
end the FOR loop before any of the commands are executed!

If you don't express the FOR loop as a single logical line of code, only the first line will
actually execute within the loop. In your case, that would be the line that reads

for i = 0, s[0]-1 do begin

The loop will increase the variable i from 0 to s[0]-1, without doing anything else.
Once that's done, the script will execute the next line (write_png ...) and write one

image for the particular case, i = s[0]-1, which is the value of i at the end of the loop.

You only need to express a FOR loop as a single line of code if
(1) you're at the command line or (2) you're creating a batch file
(e.g. mybatchfile.pro) and running it using the '@' directive (e.g. IDL> @mybatchfile).

All the best,

David