Subject: idl print call
Posted by audrey.schaufelberger on Sun, 26 Mar 2017 12:32:01 GMT
View Forum Message <> Reply to Message

Hey!

I am running an IDL procedure that is called several times from a bash script, each time with varying parameters. The output of each call is written to a separate file.

Is it possible to write at the beginning of the output file the call itself? I use the time stamp in the file name to discern the different calls, but it is annoying to check the time stamps each time to know which file belonges to which call...

best,
Audrey

Subject: Re: idl print call
Posted by Markus Schmassmann on Mon, 27 Mar 2017 11:26:31 GMT
View Forum Message <> Reply to Message

On 03/26/2017 02:32 PM, audrey.schaufelberger@gmail.com wrote:
> I am running an IDL procedure that is called several times from a
> bash script, each time with varying parameters. The output of each
> call is written to a separate file.
>
> Is it possible to write at the beginning of the output file the call
> itself? I use the time stamp in the file name to discern the
> different calls, but it is annoying to check the time stamps each
> time to know which file belonges to which call...
Hi Audrey

I don't think there is a general way to do retrieve the file call, some foreknowledge is required.
From within IDL it is probably not possible to know how you capitalized the name of the procedure you started, whether you passed your arguments by e.g. '-arg aaa -arg bbb' or '-args aaa bbb', and in general it won't be possible to know what your arguments were before shell expansion. Furthermore, IDL won't know whether you nice'd the call or whether you piped the output to a log, and some other things I probably missed.

Knowing what your bash script does should make it rather easy to construct the call from the list of arguments.
How to get to the list of arguments should become clear when reading the procedure below, some easy examples of probable calls are included too.
Should you really need the actual call without using any foreknowledge, use
spawn, 'history', out
and then parse out .

I hope that helps, Markus

for demonstration purposes enter in bash:

```
idl -e procedure_fromBash -args test one two
idl -e "procedure_fromBash, ['test','one','two']"
echo "test" > args4IDL
echo "one" >> args4IDL
echo "two" >> args4IDL
idl -e procedure_fromBash
rm args4IDL
```

```
;;; file procedure_frombash.pro ;;;-----------------------------------
pro procedure_fromBASH, argsFromBash
    n=n_elements(argsFromBash)
    if n eq 0 then print, $
       'procedure_fromBASH has not been passed an argument' else begin
       print, 'procedure_fromBASH has as argument a '
       help, argsFromBash
       print, 'which contains:'
       for i=0,n-1 do print, argsFromBash[i]
       print, 'on one line this is:'
       print, strjoin(argsFromBash,' ')
       print, 'the call was probably equivalent to'
       print, 'idl -e "'+"procedure_fromBash, ['"+$
          strjoin(argsFromBash,"', '")+"']"+""
    endelse

    argsCL=command_line_args(count=nargs)
    if nargs eq 0 then print, $
       'no command line argument has been passed' else begin
       print, 'command line argument is'
       help, argsCL
       print, 'which contains'
       for i=0,nargs-1 do print, argsCL[i]
       print, 'on one line this is:'
       print, strjoin(argsCL,' ')
       print, 'the call was probably equivalent to:'
       print, 'idl -e procedure_fromBash -args '+strjoin(argsCL,' ')
       print, 'see http://www.harrisgeospatial.com'+$
          '/docs/COMMAND_LINE_ARGS.html'
    endelse

    if ~file_test('args4IDL') then print, $
       "file 'args4IDL' does not exist" else begin
```

```
      spawn, "wc -l args4IDL | awk '{print $1}'", out
      nlines=fix(out)
      if n eq 0 then print, "file 'args4IDL' is empty" else begin
         argsFromFile=strarr(nlines)
         openr, lun, 'args4IDL', /getlun
         readu, lun, argsFromFile
         free_lun, lun
         print, 'argument read from file is'
         help, argsFromFile
         print, 'which contains:'
         for i=0,nlines-1 do print, argsFromFile[i]
         print, 'the call was probably equivalent to:'
         print, 'idl -e procedure_fromBash'
         print, "with file 'args4IDL' containing:"
         for i=0,nlines-1 do print, argsFromFile[i]
      endelse
   endelse

   case 1 of
      n      gt 0:    args=argsFromBash
      nargs  gt 0:    args=argsCL
      nlines gt 0:    args=argsFromFile
      else       :    args=!null
   endcase
end
```

Subject: Re: idl print call
Posted by Markus Schmassmann on Mon, 27 Mar 2017 12:15:13 GMT
View Forum Message <> Reply to Message

On 03/26/2017 02:32 PM, audrey.schaufelberger@gmail.com wrote:
> I am running an IDL procedure that is called several times from a
> bash script, each time with varying parameters. The output of each
> call is written to a separate file.
>
> Is it possible to write at the beginning of the output file the call
> itself? I use the time stamp in the file name to discern the
> different calls, but it is annoying to check the time stamps each
> time to know which file belonges to which call...
or for short:

```
spawn, 'ps -p $(ps -p $$ -o ppid --no-heading) -o cmd --no-heading', out
print, out
```

see:
https://ss64.com/bash/ps.html
