SESUG Paper 139-2020

Executing long-running processes under UNIX/Linux

David B. Horvath, MS, CCP

ABSTRACT

While Enterprise Guide and Studio are great tools, they require continued connections while your process is running. If the connection ends for any reason (network, VPN, local power outage, etc.), you lose the results. By executing Grid SAS® from the operating system command line, so long as the server remains up, you will be able to get the results.

Topics:

- Logging onto the server
- Moving your code to the server (via SFTP)
- Executing your SAS code in the background
- Monitoring execution
- Viewing/Retrieving logs
- Basics of vi (visual editor).

INTRODUCTION

In this time of COVID-19, many of us are working remotely and often over a VPN. The added steps in those connections increases the opportunities for failures and disconnects. While that is annoying during the development process, it becomes a significant issue when you have long running processes.

Fortunately, there are techniques for kicking off execution on the server itself that will continue even if the user connection is severed.

Many organizations enforce timeouts on VPN and server access – either as a security measure or as a means of sharing limited resources. Some are based purely on activity (commands executed, keys struck, data transferred between SAS EG and the server) while others are purely time based (24 hour limit on VPN connections).

The solution is the use of the command line directly on the server. This is also known as “the shell” or “shell access” and is accessed via application software such as “BlueZone” or “putty” or other terminal emulators.

TERMINAL EMULATOR

First you’ll need to select the terminal emulator and install it. The choice may depend on your organization and internal standards.

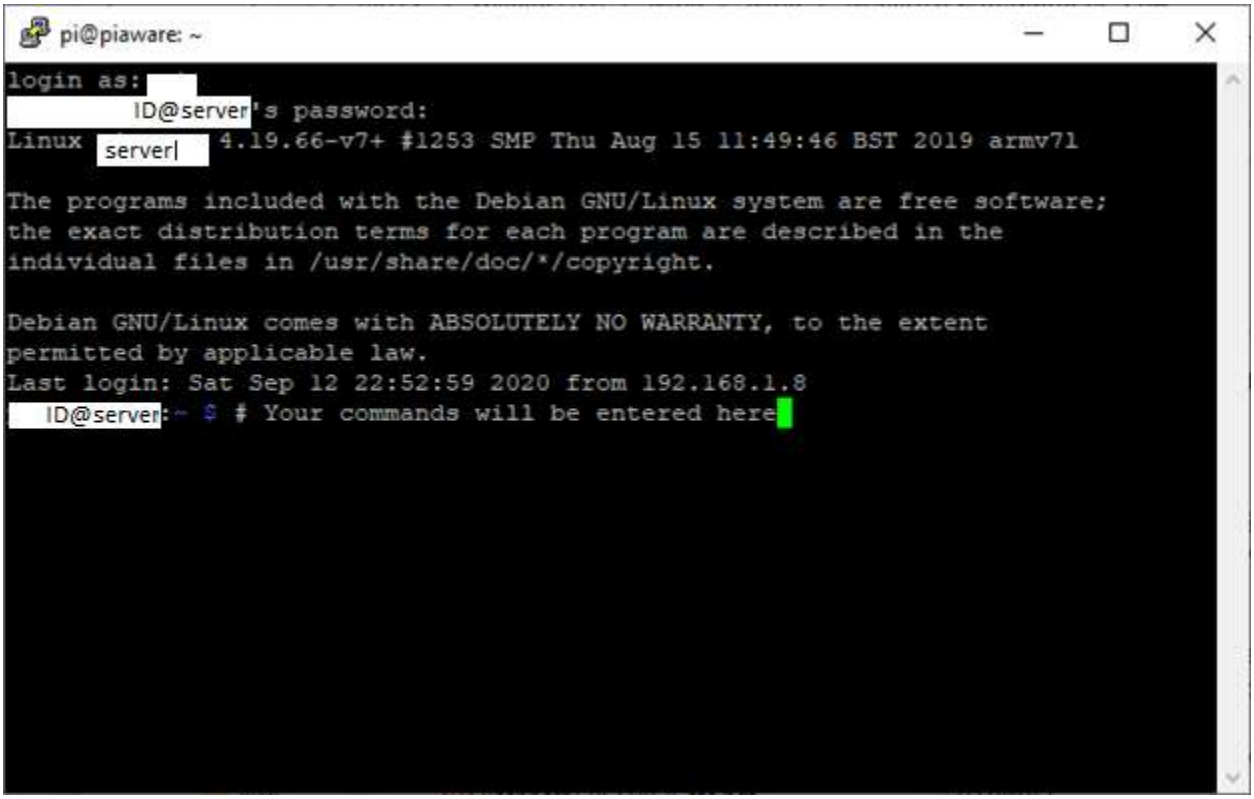
Once the software is installed, you will have to configure it with the name of the server to access along with a port number (typically 22). Some of the tools provide a mechanism to keep the connection alive. I tend to use “NOP” (no operation) so that it does not interfere with my typing with a cycle time of 1 minute (low load on connection and server).

But you will have to research the settings for your situation.

GETTING STARTED IN LINUX

After starting the terminal emulator and starting the connection to your server, you will be prompted for user ID and password.

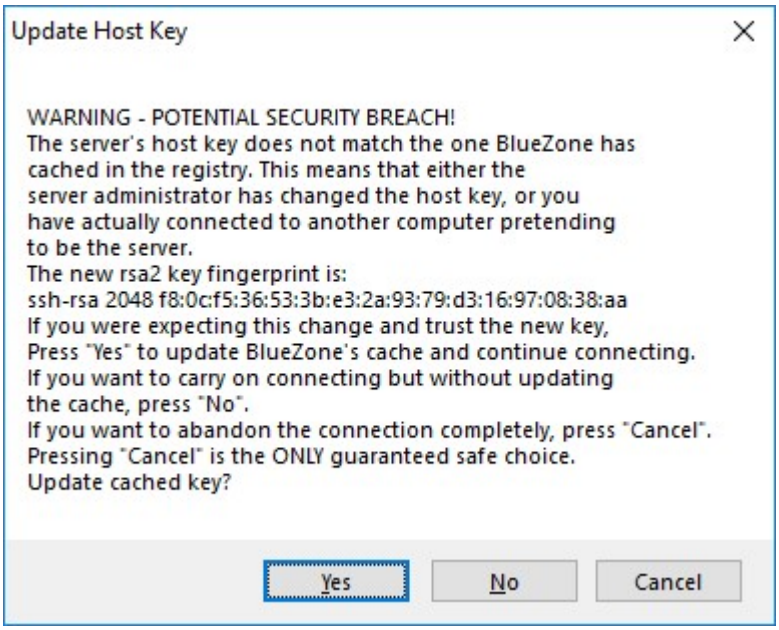
Display 1 Logging into UNIX/Linux.



Display 1. Logging into UNIX/Linux

You may receive a warning like the following. Generally, you can safely ignore these messages the first time you connect to a server or if it is on a load balancer (you use one name but might get directed to one of the several servers with the lowest load).

Display 1 Common Warning when logging in.



Display 2. Common Warning when logging in

IMPORTANT COMMANDS

By default, you start in your "home" directory (much like a Windows C: drive). Depending on your organization, you may be directed to store your files (including programs) in another directory. You will have to ask locally to determine that.

If your organization has server timeouts enabled, you should enter the following command to disable the timeout:

```
export TMOUT=0
```

If you have been directed to store your files in another directory, you will also need to enter the following command (modified to point to the appropriate directory):

```
cd /the/path/you/were/assigned
```

Some other useful commands:

- cd - – previous location (toggle)
- df -h . – display available (and used) space for current location
- pwd – show your current directory
- ls – list (show) directory contents
- ls -al – list all directory contents, long format
- # - anything that follows is a comment

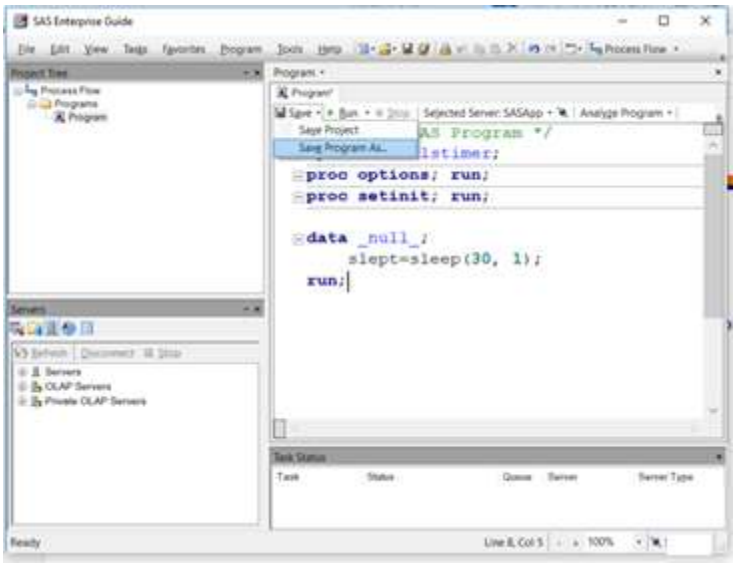
EDITING YOUR CODE

There are two places you can edit your code

- On the PC (SAS EG or other editor)
- On the Server (vi or emacs)

For simplicity, I'm going to focus on use of the PC for editing, server for execution. You really have two alternatives. First, you can save your file directly from SAS Enterprise Guide to the server. Or you can use SFTP to move your SAS code. Either way, you have to save your program as a separate SAS file and not part of the Project. This is done through the “Save Program As” option in the Program window of EG. If you’re using another editor, you’ll have other choices.

Display 1 Saving Code

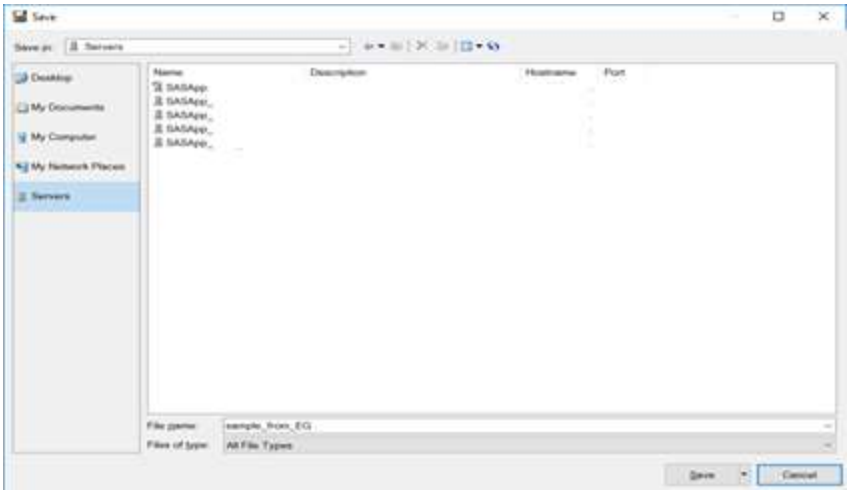


Display 3. Saving Code

SAVING YOUR CODE TO THE SERVER DIRECTLY FROM SAS EG

The easy way is to allow EG to write the file to the server directly by selecting Servers on the left side and the appropriate server (likely labeled as SASApp):

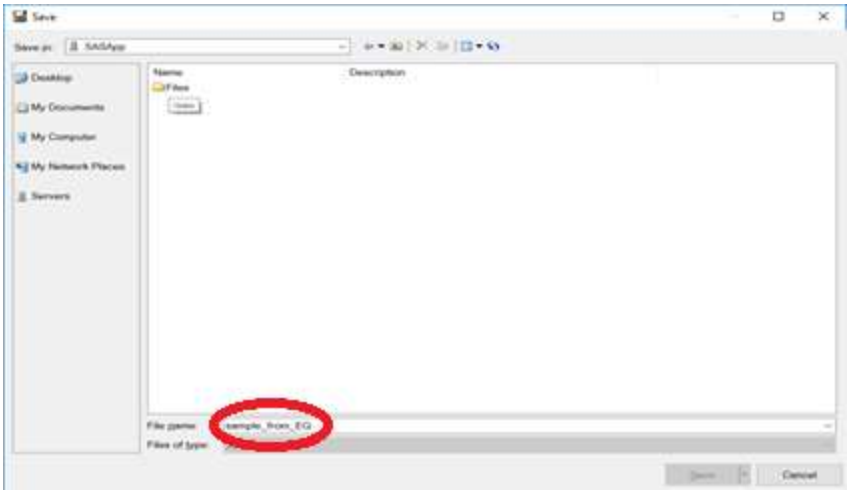
Display 1 Select the Server



Display 4. Select the Server

From there, you will navigate to the appropriate directory (like you did with the 'cd' command when you logged onto the command line environment. Once you get to the proper location you'll enter your preferred file name (circled in red below) and click on the Save button. If Save is grayed out, you likely do not have permission to write into that directory.

Display 1 Enter the Name



Display 5. Enter the name

I can also go into the command line environment to review the file by entering the 'ls' command::

```
[ID@Server]$ ls -al
-rwxrwx---  1 ID GROUP_ID         204 Apr 29 11:12 sample_from_EG.sas
```

SFTP OF SAS FILE FROM WINDOWS TO SERVER

We can also save the SAS program file to Windows and use SFTP: Secure File Transfer Protocol to move the files to the server. There are GUI and command line versions available. After you start the software, you will be prompted for server, ID, and password. You may even see the "POTENTIAL SECURITY BREACH!" warning.

Once you have connected, you'll need to navigate to the appropriate directories – Windows and Server.

Hopefully, you will be using a GUI tool because all you'll need to do is drag and drop the file between windows. Command line tools are more complex and beyond the scope of this paper

EXECUTING ON THE SERVER

Once the file is on the server, return to the command line. You need to make sure the code was transferred (or that you are in the proper directory). I don't want to admit the number of times I have been looking for a file, am unable to figure out why it is not in the proper place, only to realize that I am in the wrong place – so, of course, the file will not be there.

As before, I can use the 'ls' command to confirm the file is there:

```
[ID@Server]$ ls -al
-rwxrwx---  1 ID GROUP_ID         204 Apr 29 11:12 sample_from_EG.sas
```

To run the program, I will use the following command (all on one line):

```
nohup  sasgsub  -gridwaitresults -gridsubmitpgm sample_from_EG.sas >
YOURINIT.out &
```

Note that the program name should be whatever you saved your program as. I recommend your first program be a really simple one like I showed – merely running proc options and proc setinit.

Let me break that line down into its component pieces:

- nohup – allow to run even if the terminal disconnects (ignore "hang up")
- sasgsub – SAS Grid Submit command
- -gridwaitresults – wait for the log to be returned
- -gridsubmitpgm – submit a SAS program (can also submit UNIX/Linux Commands)
- sample_from_EG.sas – your SAS program name
- > YOURINIT.out – save the command output to YOURINIT.out
- & - return control to the terminal for more commands while this runs

An actual run may look something like the following:

```
[ID@Server]]$ nohup sasgsub  -gridwaitresults -gridsubmitpgm sample_from_EG.sas > YOURINIT.out &
[1] 109981
[ID@Server]$ nohup: ignoring input and redirecting stderr to stdout
```

"[1] 109981" is the job number and process ID on the current server. There is also a remote job executing within the grid with a separate tracking. We can see what is running on the grid through the 'bjobs' command:

```
[ID@Server]$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
91088   ID      RUN   normal     SERVER      GRID01      *_20200419 Apr 19 09:23
92278   ID      RUN   normal     SERVER      GRID02      *e_from_EG Apr 19 14:29
```

I see that my EG session is still active (first line) as well as the command I submitted (second).

In addition, I still have the process I submitted locally which I can see with the 'jobs' command:

```
[ID@Server]$ jobs
[1]+  Running                 nohup sasgsub -gridwaitresults -gridsubmitpgm sample_from_EG.sas >
YOURINIT.out &
```

At this point you can log off the server using 'logout' or 'exit' – your processes will keep running until complete (or they encounter an error). Or if you want to monitor progress, hitting enter periodically will eventually result in:

```
[ID@Server]
[1]+  Done                    nohup sasgsub -gridwaitresults -gridsubmitpgm sample_from_EG.sas >
YOURINIT.out
```

Note that you can have multiple processes executing at the same time. Each local background process will have a unique job number within square brackets [].

LOOKING AT EXECUTION RESULTS

Once the execution has completed, we can look at the results. First look at the directory contents:

```
[ID@Server]$ ls -l
-rwxrwx---  1 ID group_id         129 Apr 19 14:10 sample_from_EG.sas
drwxrws---  2 ID group_id      131072 Apr 19 14:28 SASGSUB-2020-04-19_14.28.01.972_sample
-rwxrwx---  1 ID group_id         784 Apr 19 14:28 YOURINIT.out
```

The results are in the `SASGSUB-2020-04-19_14.28.01.972_sample` directory. There will be .log and .lst files (as appropriate to your process). You can look at them on the server using the 'more' command to page

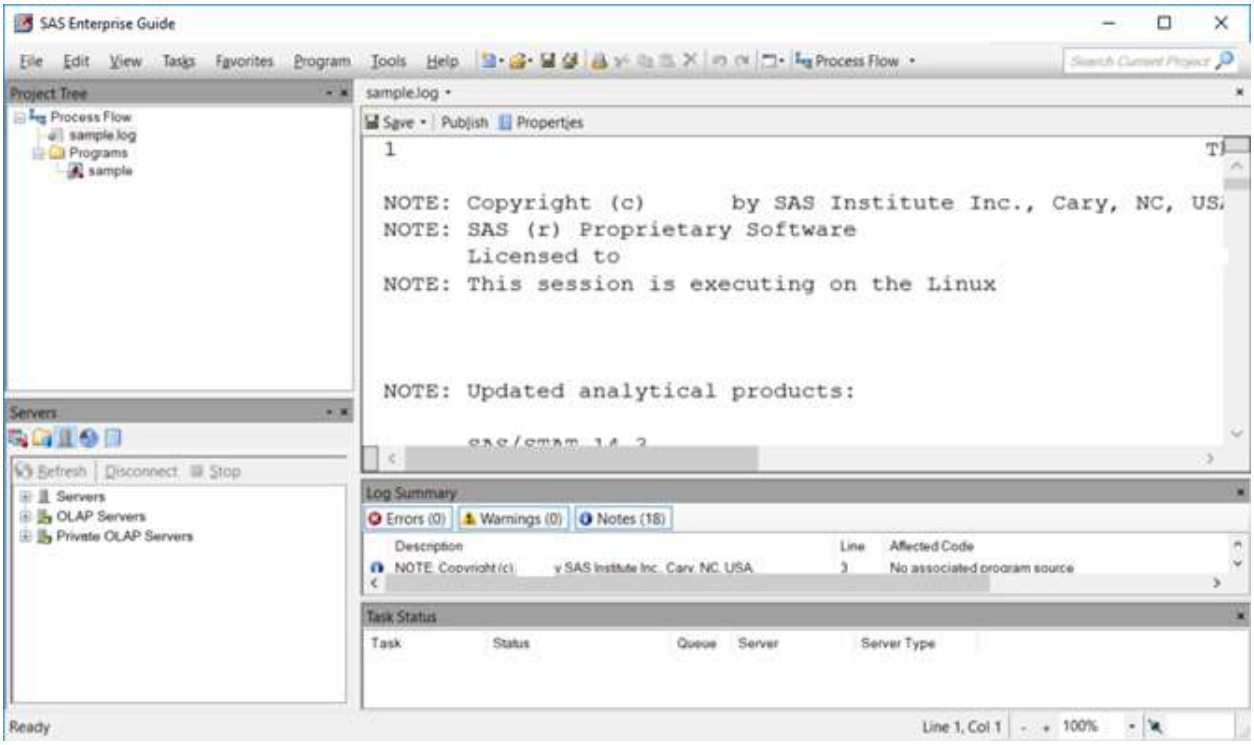
through (pressing the spacebar to get the next screen, ‘q’ to quit) or using the ‘cat’ command to see the entire file. You can even transfer the file back to Windows to make use of tools there.

Because of the time involved, your earlier SFTP session has likely timed out so you will need to repeat the steps to get to the proper directory. Then you can drag and drop the log file to a Windows directory.

Once the log is a file on your PC that you can open in SAS EG by clicking on File, Open, Other, and select the appropriate file.

Then you can scroll or search through your log

Display 1 Log Results



Display 6. Log Results

STATUS AND KILLING PROCESSES

After submitting a process, it is possible to check status of your grid jobs using the –gridgetstatus option of the ‘sasgsub’ command:

```
[ID@Server]$ sasgsub -gridgetstatus all
SAS Grid Manager Client Utility Version 9.45 (build date: Oct  3 2017)
Copyright (C) 2009-2017, SAS Institute Inc., Cary, NC, USA. All Rights Reserved
  922759 (sample) is Finished:  Submitted: 03Apr2020:14:22:09, Started: 03Apr2020:14:22:10 on
Host lsae331a, Ended: 03Apr2020:14:22:42, RC:0
  922872 (sample) is Submitted: Submitted: 03Apr2020:15:00:03
 [ID@Server]$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
91088   ID      RUN   normal     Server      Grid04      *_20200419 Apr 19 09:24
92287   ID      PEND  normal     Server                  sample     Apr 19 15:00
```

To cancel an executing (or pending) process, you can use the –gridkilljob option of the ‘sasgsub’ command. Although there is a ‘bkill’ command, please do not use it. I have seen cases where the

‘sasgsub’ command will get stuck in a very tight loop (actually tying up an entire CPU core) when ‘bkill’ was used to terminate a SAS Grid process:

```
[ID@Server]$ sasgsub -gridkilljob 92287
SAS Grid Manager Client Utility Version 9.45 (build date: Oct  3 2017)
Copyright (C) 2009-2017, SAS Institute Inc., Cary, NC, USA. All Rights Reserved
Termination requested for job ID 922872.
```

The following line in YOURINIT.out tells you it was killed:

```
Grid job return code is 255 (0xff)
```

And in the SAS Log:

```
ERROR: User asked for termination
NOTE: The SAS System stopped processing this step because of errors.
```

BASICS OF VI

If you would rather not switch between Windows and UNIX/Linux repeatedly (using Windows to edit your code), you can use the ‘vi’ command (Visual Editor) which is one of the primary text/programming editors under UNIX/Linux. I will caution you that the command syntax is a bit complex but it has a huge advantage – it is Is available on just about every platform (from Android Phone to MAC OSX to Windows to Super Servers running Linux).

To invoke the editor, you would use one of the following commands. It will create the file if it does not already exist. In read-only mode it will not allow you to save any changes:

- vi sample.sas – allows editing
- vi -r sample.sas – read-only mode
- view sample.sas – also read-only mode

The editor actually has three internal modes:

- Command – default allows you to move around
- Line – allows entry of 'ex' commands (like global search and replace)
- Insert – entering data

Some key commands:

- <esc> gets you from Input to Command Mode
- :q! forces exit (even if you've changed the file)
- Ctrl-f page forward
- Ctrl-b page back
- 1G – go to line 1
- G – go to last line
- Ctrl-g location in file
- /string – find string going forward, n – find next down
- ?string – find string going backward, n – find next up
- :set ic <enter> – ignore case when searching

There are many other commands but there are entire books written about using ‘vi’.

CONCLUSION

Running your SAS code directly on the server removes many of the causes of unexpected termination (due to being disconnected). Although running on the server is more complex and requires some additional learning, it means that your long processes get to complete rather than dying repeatedly. This is especially important with significant remote work occurring due to COVID-19.

ACKNOWLEDGMENTS

I want to thank the organizers of this great conference – I would have preferred to present in person but certainly understand the need for the conference to be paper-only. I will miss the interaction with other speakers and attendees. I also want to thank my employer for their past willingness to allow me to expand my horizons by attending SESUG. Lastly but certainly not least, I want to thank my spouse Mary who doesn't complain when I spend so much time at the keyboard working on documents like this.

REFERENCES

SAS Grid Submit Command available at https://documentation.sas.com/?docsetId=gridref&docsetTarget=n0l4rpkteaappxn1g0vuj567dajj.htm&docsetVersion=9.4&locale=en

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

David B. Horvath, CCP
+1-610-859-8826
dhorvath@cobs.com
http://www.cobs.com
LinkedIn: https://www.linkedin.com/in/dbhorvath/