

Paper 4952-2020**SAS® Options Precedence in SAS® Grid Environment**

Piyush Singh, Sumit Bhati, Lavanya Erisetti
TATA Consultancy Services Ltd., Indianapolis, IN.

ABSTRACT

This paper helps to understand the precedence of SAS® Options mentioned in different configuration files. There are multiple configuration scripts files that get deployed with SAS® Grid install, and this paper explains the purpose of these scripts and how we should edit these scripts to meet the new business needs for SAS® options which are not present with default SAS® Grid deployment. The codes and techniques given in this paper will describe how to edit these configuration scripts files to set the desired options or environment variables.

There can be different kinds of jobs to be executed in SAS® Grid. The Platform Administrator creates different SAS® Application Server context for different kinds of business requirements or different user groups. It becomes important to know how these different SAS® Application Servers are being used (with different SAS® jobs). This paper explains how we can use the SAS® configuration files to get such information in an easy way by understanding the precedence of SAS® options in SAS® Grid platform.

INTRODUCTION

The SAS Grid is a complex deployment in comparison to stand alone SAS environment. There can be multiple tiers in SAS Grid like SAS Metadata Server, Control Server, and Stored Process Server. These tiers can use multiple SAS configuration files with different sets of SAS options. One needs to find the appropriate configuration files to set the required SAS options for specific SAS clients. In particular, this paper explains the SAS options for two SAS clients, SAS Enterprise Guide and SASGSUB.

SAS GRID TIERS AND HIERARCHY

Below are the key different tiers in SAS Grid environment and each of the tiers have their own config file as per the architecture:

SHARED SAS BINARIES WITH SAS APPLICATION SERVERS CONTEXTS

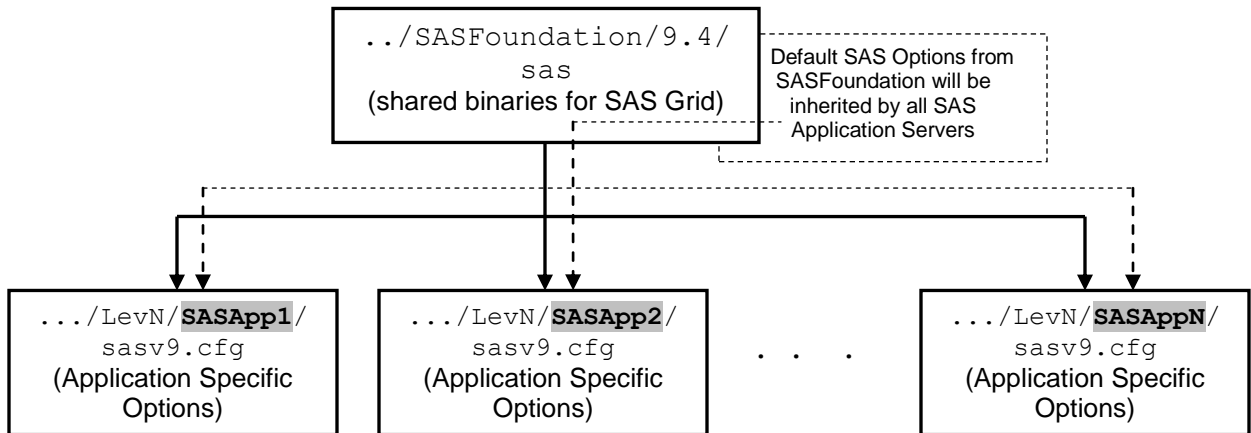
The key concept of SAS Grid is to use shared binaries with all of the compute nodes and load share among all the compute nodes. Having the shared file system, it helps to use one deployment and create as many application servers as you need. Different SAS Application servers work almost like different SAS servers without doing multiple install of SAS software.

Shared deployment goes through <ControlServer>/sashome/SASFoundation/9.x and you can use the options and configuration defined to this level. If you define SAS options in the configuration file in <ControlServer>/sashome/SASFoundation/9.x, this can be used in all the SAS Application servers defined in the SAS Grid environment using this shared sashome.

The diagram below (fig-) shows how you can use shared deployment of SASFoundation/9.x between different SAS Application servers, and if you need to define a specific set of SAS options for each SAS Application server, you can define those SAS options in the respective

SAS Application configuration file, and the default SAS options will be inherited from the SAS Foundation configuration file.

Figure 1. Multiple SAS Application Servers in SAS Grid



MULTIPLE SAS SERVERS WITHIN SAS APPLICATION SERVER

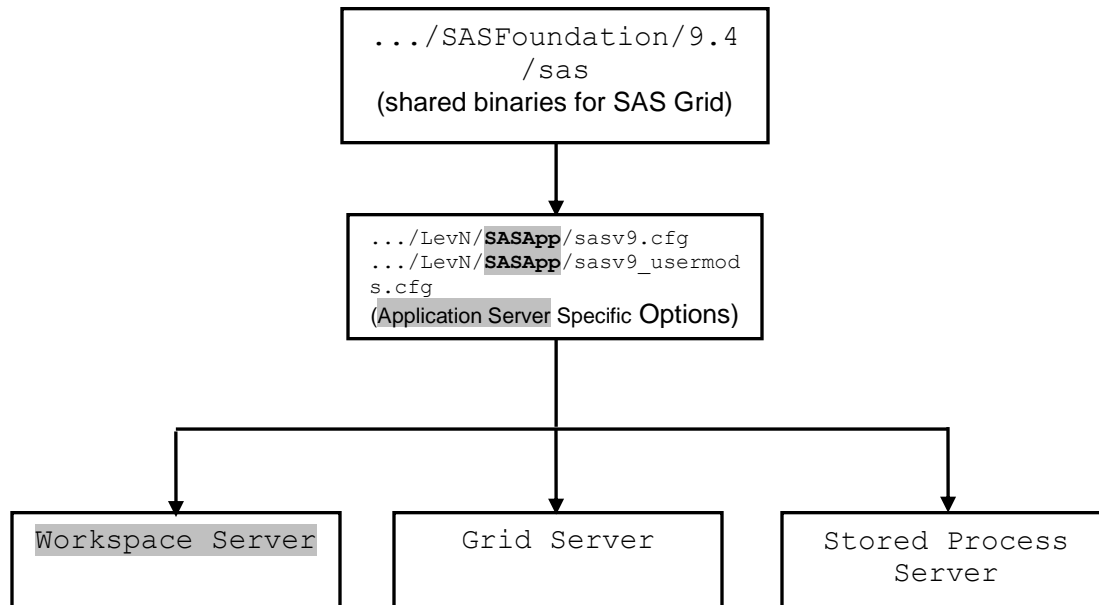
After deploying the SAS Foundation for Grid, we need to configure the SAS Application servers for different kinds of SAS Clients like SAS Enterprise Guide, SASGSUB, and others. It's not necessary that all SAS Application servers should be configured for all SAS clients, and you can configure as per the need and requirements. The diagram below shows the three SAS clients for the SAS Application server. We split the different Application server using the same SAS Foundation; in this way, all the SAS clients can use the shared configuration file like `sasv9_usermods.cfg` in the specific SAS Application server (`.../sasconfig/LevN/SASApp/sasv9_usermods.cfg`) and if you need to define some additional SAS options for a particular SAS client, then you need to go inside the SAS client's directory and define it there in SAS configuration file.

For example, in the diagram below, if you need to define some specific SAS option for the SAS Enterprise Guide client for the server "SASApp" (gray highlighted), then we need to define the SAS options in the configuration file that resides in the "Workspace Server" directory (Linux) like this

`.../sasconfig/LevN/SASApp/WorkspaceServer/sasv9_usermods.cfg`. Since it's the Workspace Server that spawns the SAS session for SAS Enterprise Guide, we need to define the SAS Enterprise Guide related options in the Workspace server configuration file (`sasv9_usermods.cfg`). It's not recommended to modify the `sasv9.cfg` file; rather you can update `sasv9_usermods.cfg` file to prevent it from any accidental issue with your original deployment.

SASGSUB doesn't use the Workspace server to launch the SAS session, and for this, there is a separate directory inside the SAS Application Server named "GridServer". If you need to modify or specify any specific settings, you can refer to the `grid_usermods.cfg` file. Along with SAS options, this file can be referred to in order to add or modify it for many other Grid related options.

Figure 2. Multiple SAS Applications in a SAS Application Server "SASApp"



THE OVERALL PRECEDENCE OF SAS OPTIONS IN THE SAS GRID ENVIRONMENT

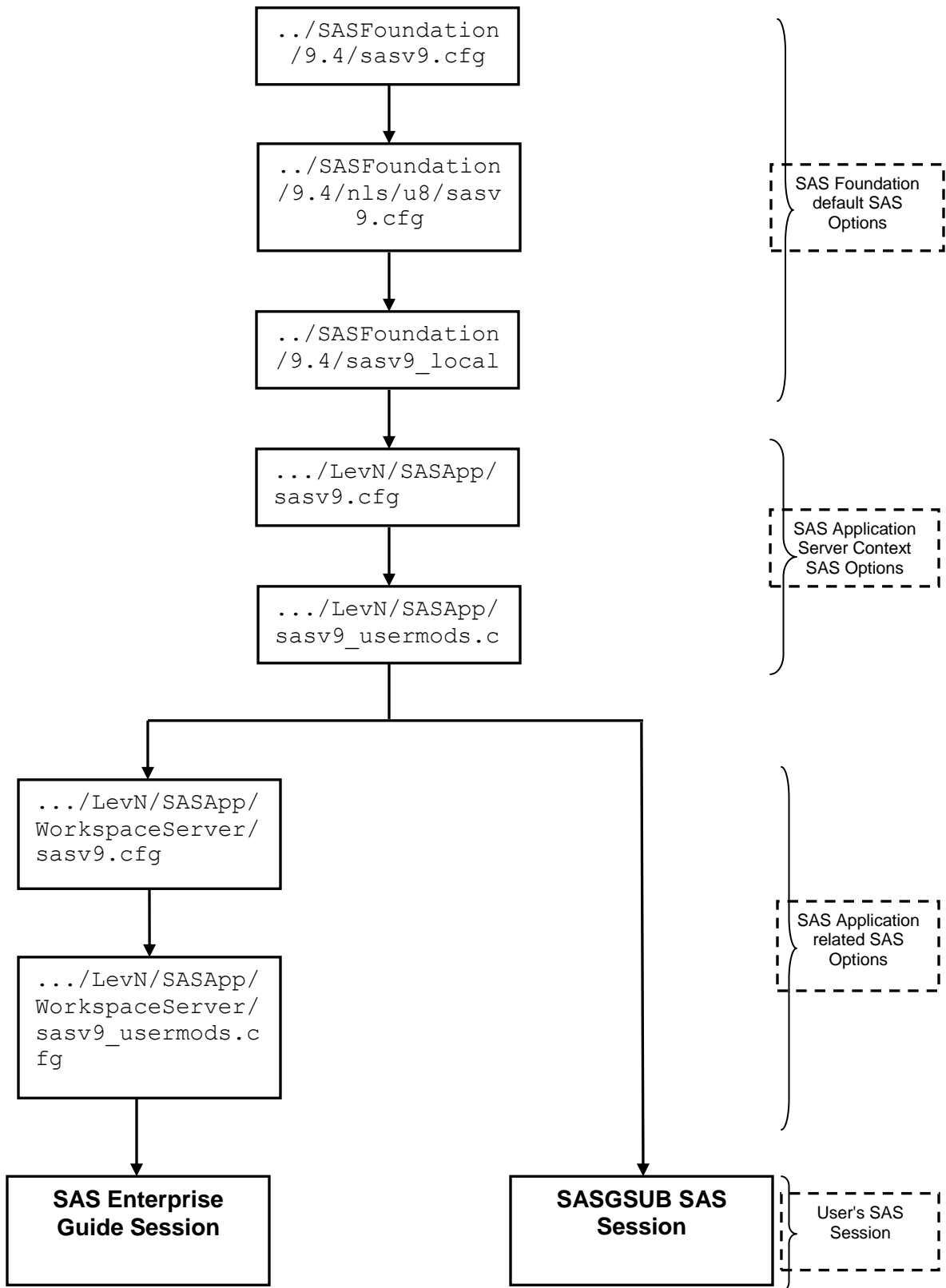
As discussed in earlier in this paper, there are two hierarchies for the SAS session from the deployment point of view. The first one comes from the shared deployment of SAS software and the default options from the deployment cascade throughout the hierarchy unless this is overridden by subsequent SAS configuration files. SAS software comes with all these automatic SAS options for the initial deployment, but these can be changed as per your business need. The second Level of SAS options comes from the default SAS Application Server Context deployment. You can create multiple SAS application server contexts to meet different business needs or for different groups of users, and you can set the SAS options for that specific business or group of users.

The above two sets of options come from the deployment side, but there you can set a few of the SAS options in your running SAS session as well. If you want to use some specific SAS options just for your SAS session, you can specify them on top of your SAS program, and those would be effective for that program only, not outside that specific SAS session. You can use the "options" statement to define your one value instead of using the default one. The statement below sets the page size and line size to a given value:

```
options pagesize=80 linesize=64;
```

But the key point is, not all SAS options can be changed in your running SAS session. A few of the options get initialized while invoking the SAS session, and if you change such options in your running SAS session, you will get the desired result. For such options, you might need to add in the deployment configuration files, or you can create your own config files or environment variables, which is explained briefly in a coming section of this paper.

Figure 3. SAS Options Precedence in SAS Grid



LIST THE CONFIGURATION FILES USED IN YOUR SAS SESSION

You can use "proc options option=config;" procedure to list the configuration files executed in your current SAS session. This will list the files in incremental precedence order. The second file takes precedence over the previous in top to bottom order. If there is an option defined in more than one of the config files, the value from the file with higher precedence will take priority. Below is the format of proc config procedure output:

```
proc options option=config;
run;
```

Partial SAS Log:

```
. . .
config=(/ <compute_home>/sashome/SASFoundation/9.4/sasv9.cfg
/ <compute_home>/sashome/SASFoundation/9.4/nls/u8/sasv9.cfg
/ <compute_home>/sasconfig/Levl/SASApp/sasv9.cfg
/ <compute_home>/sasconfig/Levl/SASApp/sasv9_usermods.cfg
/ <compute_home>/sasconfig/Levl/SASApp/WorkspaceServer/sasv9.cfg )
. . .
```

In the above sequence "...sasconfig/Levl/SASApp/WorkspaceServer/sasv9.cfg" is the last configuration file and has highest precedence, and if a SAS option is defined in workspaceserver config file, it will take precedence if this was defined in any of the earlier configuration files.

But the listing from "proc options" procedure shows the config file precedence for that particular SAS session only. For example, if you run the same procedure for different SAS clients in the same SAS Grid environment, output may be completely different because of the time it takes the SAS session to connect to a specific SAS Application. So, if you need to set SAS options for a specific SAS application, you can run the procedure to that specific SAS application to find the appropriate configuration which should be modified to accommodate the changes. This provides the flexibility for SAS Grid users to use different sets of SAS options with each of the SAS Application servers and different options with SAS Applications within the same SAS Application Server Context.

OTHER WAYS TO CHANGE THE SAS OPTIONS

The techniques in previous sections (default deployment with shared binaries and SAS Application Server Context) are the standard way to change and read the SAS options, but there are other ways as well to use your values for SAS options; a few of them are given below:

CREATE YOUR OWN CONFIGURATION FILE

If you are savvy SAS user and don't want to rely on SAS administrator all the times, you can create your own configuration file and use for yourself. This will be effective for your session only. Or if there are few users who need some specific options which may not be commonly used for others, you create such configuration file and place in their home directory. When user will launch SAS session, this config will be read automatically. There may be different name for this file like:

- .sasv9.cfg in your home directory (leading period makes it hidden file).
- sasv9.cfg in your home directory.

*** But this may be good fit for batch processing only, and WorkspaceServer may not use this config file. While you are running SASGSUB, it reads these user specific config files from the user's home directory without mentioning anything in your SAS program. Just having these files in your home directory is sufficient for SASGSUB to use them.

AUTOEXEC FILE

Autoexec file is being used for different purposes for a long time. You can use the SAS statements which will be executed automatically when you invoke SAS session. Like other SAS statements you can use your SAS options as well. As shown below, when you can use the OPTIONS statement, use the required SAS options:

```
options pagesize=72;
```

Unlike configuration files, the autoexec file contains both SAS options and SAS statements. The Autoexec file gets processed immediately after the launch of the SAS session.

PROVIDE OPTIONS WHILE INVOKING SAS COMMANDLINE

Along with adding SAS options in different files, you can pass the SAS options while launching SAS from the command line. This doesn't require you to save anything in any file, and as per your requirement, you can pass the SAS options for the specific job or SAS session.

```
sas -sasoption1 -sasoption2 ...
```

CONCLUSION

There are several different methods for defining the SAS options and how to set the priority. This paper provides an overview of the most commonly used methods and hierarchy of different configuration files. The intent of this paper is to help SAS users in understanding the different SAS configuration files and corresponding precedence to set the SAS options from respective SAS configuration files.

REFERENCES

Piyush Singh, Prasoon Sangwan, Ghiyasuddin Mohammed Faraz Khan. "Automation of Linux Multi-Tiered SAS® and Load Sharing Facility (LFS) Services" In *SAS Global Forum*. 2018. <https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/2757-2018.pdf>

Singh, Piyush, Sangwan, Prasoon, and Randolph, Steven. "Read SAS® Metadata in SAS® Enterprise Guide" In *SAS Global Forum*. 2017. <http://support.sas.com/resources/papers/proceedings17/1275-2017.pdf>

Sangwan, Prasoon, Tanuj Gupta, and Piyush Singh. "Key Requirements for SAS® Grid Users." In *SAS Global Forum*. 2016. <https://support.sas.com/resources/papers/proceedings16/7140-2016.pdf>

Piyush Singh. "Enhancing SAS® Piping Through Dynamic Port Allocation" In *SAS Global Forum*. 2014. <http://support.sas.com/resources/papers/proceedings14/1826-2014.pdf>

RECOMMENDED READING

- *SAS® 9.4 Intelligence Platform: Application Server Administration Guide*
- *SAS® 9.4 Companion for UNIX Environments, Sixth Edition*
- *SAS® 9.4 System Options: Reference, Fifth Edition*

CONTACT INFORMATION

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

Piyush Singh

piyushkumar.singh@tcs.com

www.tcs.com

Sumit Bhati

sumit.bhati@tcs.com

www.tcs.com

Lavanya Erisetti

lavanya.eerisetti@tcs.com

www.tcs.com