

Paper 470-2013

The Top Four User-Requested Grid Features Delivered with SAS® Grid Manager 9.4

Doug Haigh, SAS Institute, Inc.

ABSTRACT

As more and more SAS® customers choose to deploy their platform for SAS® Business Analytics with SAS® Grid Manager, we continue to expand the capabilities of a shared, managed, and highly available environment. Several features have been added in SAS 9.4, providing the following:

- easier administration of the different types of SAS users in the grid
- increased management of more SAS components running in the grid
- integration of the grid with IT standards, such as an existing enterprise scheduler
- increased debugging capabilities for quick problem resolution

This paper details how to accomplish all of the above through new grid options sets, workspace servers spawned on the grid, new options in SAS Grid Manager Client Utility, and enhanced logging.

INTRODUCTION

In SAS 9.4, SAS Grid Manager has added new capabilities to make both a SAS administrator's and grid administrator's lives less complicated. SAS Grid Manager has added the ability to group SAS options, grid options, and grid-required resources into a "grid options set" that can be associated with a specific application for a specific identity under a single SAS Application Server. The SAS Object Spawner has expanded its grid capabilities by leveraging the grid to control the starting of workspace servers. The SAS Grid Manager Client Utility will now be able to submit a SAS job to the grid, wait for it to complete, and grab the results in a single invocation so that it can be better used by third-party scheduling software. Lastly, the debugging capabilities provided by the SAS logging facility have been incorporated into all the parts of the SAS Grid Manager product, which makes problem determination quicker and easier.

GRID OPTIONS SETS

SAS administrators face a decision when deploying a grid about how to manage the applications that users or groups will use on the grid. Grid policy is typically managed by creating a queue for each group or application and ensuring that when users start using the grid, their work will be routed through the proper queue. You can have queues for batch processing, interactive processing, and nightly processing just to name a few. Even when a job is submitted to the correct queue, it might want to run on a subset of machines that meet a special resource requirement and use some special SAS options to take advantage of that resource.

We have heard from many customers that they would like to specify a set of options that would be used by the grid jobs and be able to map those options to a user or group for a specific application without having to create a new SAS Application Server context every time a new option is required. In addition, they want this to be specified in metadata and completely transparent to the SAS Enterprise Guide, SAS Enterprise Miner, and SAS Data Integration Studio users.

In SAS Grid Manager 9.4, we have added this capability in what we call a "grid options set." A grid options set consists of a set of grid options, SAS options, and required grid resources that can be assigned when a specific application is used by a user or group. The options defined in this grid options set will be combined with any options that are defined in the logical grid server for a specific SAS Application Server, possibly overriding the default values.

This is probably best illustrated by a simple example.

Acme Widgets – a long time SAS customer – is planning to upgrade to SAS 9.4, including SAS Grid Manager 9.4. They have a Finance Department and a Risk Department that will use both SAS Data Integration Studio 4.7 to interactively create SAS programs and SAS Grid Manager Client Utility (SASGSUB) to run other SAS programs in batch mode. Acme Widgets has libraries defined in metadata, but does not want the libraries used by the Finance Department to be used by the Risk Department and vice versa.

When they originally installed and configured SAS 9.3, Acme Widgets had enlisted the help of SAS Professional Services to design their deployment. SAS Professional Services had recommended the following:

1. A SAS Application Server named “FinanceDIS” that was used when the Finance Department used SAS Data Integration Studio. It contained a logical grid server that had grid options to submit SAS grid jobs to the “priority” queue.
2. A SAS Application Server named “FinanceGSUB” that was used when the Finance Department used SASGSUB. It contained a logical grid server that had grid options to submit SAS grid jobs to the “batch” queue.
3. The Finance Department’s predefined libraries were duplicated in both the “FinanceDIS” and “FinanceGSUB” application servers.
4. A SAS Application Server named “RiskDIS” that was used when the Risk Department used SAS Data Integration Studio. It contained a logical grid server that had grid options to submit SAS grid jobs to the “priority” queue.
5. A SAS Application Server named “RiskGSUB” that was used when the Risk Department used SASGSUB. It contained a logical grid server that had grid options to submit SAS grid jobs to the “batch” queue.
6. The Risk Department’s predefined libraries were duplicated in both the “RiskDIS” and “RiskGSUB” application servers.

While not ideal, especially the duplication of predefined libraries, this setup worked well. If Acme Widgets had many more departments or more grid applications, this design would quickly grow the number of SAS Application Servers and increase the effort required to manage their metadata.

Now in SAS Grid Manager 9.4, grid options sets simplify the deployment to the following:

1. Create one application server for each department.
 - a. A SAS Application Server named “Finance.” All of the Finance Department’s libraries are assigned to it.
 - b. A SAS Application Server named “Risk.” All of the Risk Department’s libraries are assigned to it.
2. Create a grid options set for each grid application.
 - a. A grid options set named “DI Studio Grid Options” that should be used for all SAS Data Integration Studio work. It has the grid option to submit the job to the “priority” queue.
 - b. A grid options set named “GSUB Grid Options” that should be used for all SAS Grid Manager Client Utility work. It has the grid option to submit the job to the “batch” queue.
3. For each SAS Application Server, map the department’s users to the application and the grid options set.
 - a. In the SAS Application Server “Finance”, the user group “Finance” will have SAS Data Integration Studio mapped to the “DI Studio Grid Options” grid options set and SAS Grid Manager Client Utility mapped to the “GSUB Grid Options” grid options set.
 - b. In the SAS Application Server “Risk”, the user group “Risk” will have SAS Data Integration Studio mapped to the “DI Studio Grid Options” grid options set and SAS Grid Manager Client Utility mapped to the “GSUB Grid Options” grid options set.

As you can see, we can now have the same grid options set used for SAS Data Integration Studio across two SAS Application Servers. If an option change is needed for all SAS Data Integration Studio jobs, it can be made in the grid options set “DI Studio Grid Options,” and all users in all departments automatically get the change. This holds true for SASGSUB options also.

Also, if Acme Widgets grows to have more departments using SAS Data Integration Studio and SASGSUB, all they will need to create is the SAS Application Server with a logical grid server that maps the applications to the existing grid options sets for the proper set of users.

CREATING GRID OPTIONS SETS

In order to support grid options sets, the Options tab in the Grid Server Properties dialog box has been revised for SAS Grid Manager 9.4. As seen in Figure 1, there are now new fields in the dialog box. There are fields for SAS Options, Required Resources, and Grid Options that are the default for any program using the logical grid server. As was done in the previous releases, the application server name is set as a required resource name by default. Also new at the bottom is the Grid Options Set Mappings table, which indicates the grid options set used for a specific application for a specific user or group for this logical grid server.

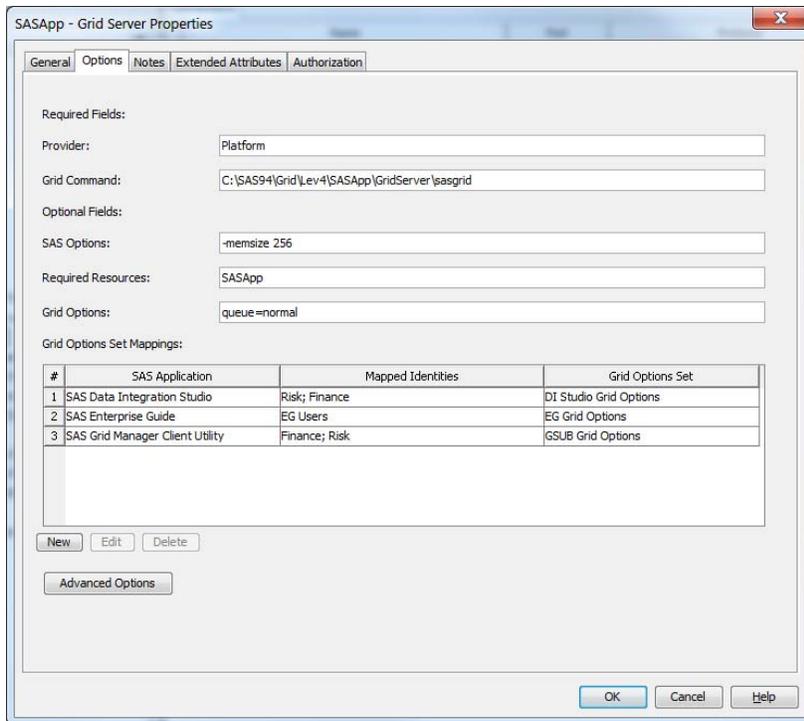


Figure 1. Grid Server Options

If the "New" or "Edit" buttons are pressed at the bottom of the table, the Grid Options Set Mapping Wizard is displayed and is similar to what is seen in Figure 2. The first field is a menu with a list of grid-capable applications. The second field has a menu with a list of the existing grid options set names. The bottom set of fields are used to select the users and/or groups that the mapping is applied to. For the dialog box shown, the application SAS Data Integration Studio will use the grid options set "DI Studio Grid Options" for the users in the Finance group and the Risk group.

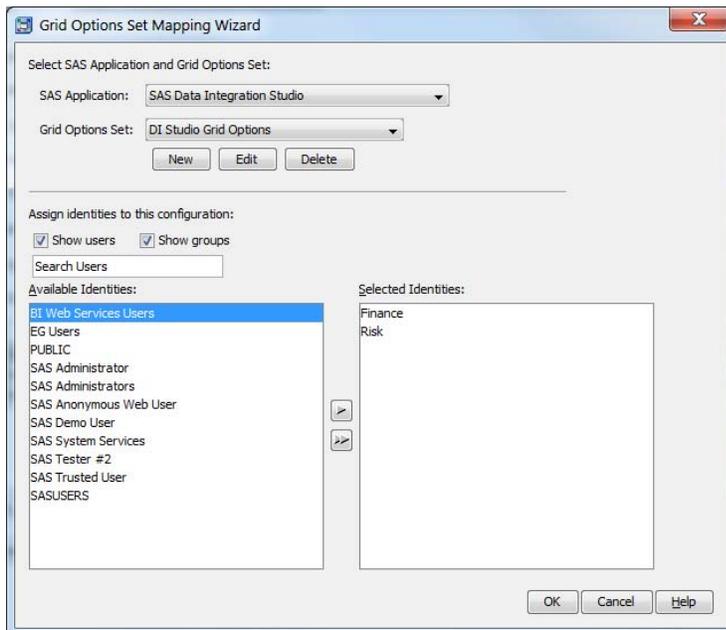


Figure 2. Grid Options Set Mapping Wizard

This dialog box is also the location to create, edit, or delete grid options sets. Use the buttons below the second field to display the dialog box used to edit the values in a grid options set as seen in Figure 3.

Figure 3. Grid Options Set Properties

This dialog box indicates that when the “DI Studio Grid Options” grid options set is used, SAS options “-memsize 0”, the required resource “MySQLDB” and the grid options “queue=priority” will be added to the default options for the logical grid server.

GRID OPTIONS SET AND DEFAULT OPTIONS

Ultimately the grid needs a set of values to use when it runs a grid job. When a grid options set is used, its option values are merged with the default option values for the logical grid server. The merging is done differently based on the type of option.

- For SAS options, the values in the grid options set are appended to the values from the default SAS Options field so that a value in the grid options set overrides any similar value found in the default SAS options field as the options are evaluated from left to right.
 - Example: The default SAS Options field has “-memsize 256” and the grid options set “DI Studio Grid Options” has a SAS Options value of “-memsize 0.” When the job is executed on the grid, the SAS command line will have “-memsize 256 -memsize 0” appended to the end of the command line. Because “-memsize 0” is evaluated after “-memsize 256” (going from left to right on the command line), it will be the memsize option used per the SAS command-line processing convention.
- For required resources, the values in the grid options set are added to the values from the default Required Resources field so that the host that a job runs on must have all values specified as a resource for the job to run on that host.
 - Example: The default Required Resources field has “SASApp” as a required resource by default and the grid options set “DI Studio Grid Options” has a Required Resources value of “MySQLDB.” When the grid provider selects a host for the grid job to run on, it will select one that has both “SASApp” and “MySQLDB” as resources associated with the host.
- For grid options, the values in the grid options set are appended to the values from the default grid options field so that a value in the grid options set overrides any similar value found in the default Grid Options field as the options are evaluated from left to right. Note: This is similar to how SAS options are processed.
 - Example: The default Grid Options field has “queue=normal” and the grid options set “DI Studio Grid Options” has a Grid Options value of “queue=priority.” When the job is passed to the SAS grid provider module, it will include the options “queue=normal queue=priority”. Because “queue=priority” is evaluated after “queue=normal” (going from left to right on the command line), the priority queue will be the queue used for the job.

So in short, SAS options and grid options in a grid options set will override any similar option from the default grid server options, whereas required resources in a grid options set are added to the required resources found in the default grid server options. This allows the grid administrator to set default values in the logical grid server fields and override or augment those values in the grid options set.

USING GRID OPTIONS SETS

To take advantage of grid options sets, a number of changes have been made to SAS products to make use of grid options sets.

SASGSUB will automatically retrieve the grid options set that has been assigned to the user specified in the metadata connection for the SAS Grid Manager Client Utility grid application. SASGSUB also has a new option “GRIDOPTSET” which can be used if there are multiple grid options sets assigned to the user.

The GRDSVC_ENABLE function has been updated to allow the SAS programmer to specify a grid options set name through the new “GRIDOPTSET” option. Applications that generate SAS code such as SAS Data Integration Studio, SAS Enterprise Guide, and SAS Enterprise Miner will be enhanced to provide a list of grid options sets that the user is allowed to use when using the specific application and then pass that option in the GRDSVC_ENABLE function.

An example of an application’s use of grid options sets is shown in Figure 4 from the Options dialog box in SAS Data Integration Studio 4.7. It shows that the grid options set name “DI Studio Grid Options” will be used for all grid-submitted code.

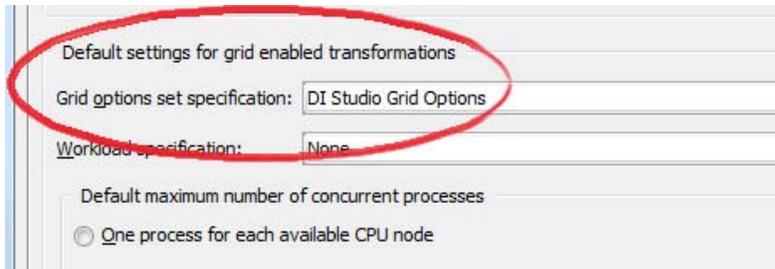


Figure 4. SAS Data Integration Studio 4.7 Grid Options Set Specification

This will cause SAS Data Integration Studio to generate a GRDSVC_ENABLE function that looks like the following:

```
grdsvc_enable(_ALL_,"server=Finance;gridoptset='DI Studio Grid Options';")
```

WORKSPACE SERVERS SPAWNED ON THE GRID

Starting with the third maintenance release of SAS 9.2, pre-code and post-code execution scripts have been provided to use in SAS Enterprise Guide and the SAS Add-In for Microsoft Office so that submitted code would run in the grid. The pre-code and post-code execution scripts started a grid server behind the scenes and routed the SAS code to the grid server for execution. This implementation allowed a policy to be applied to a user’s SAS code execution, but it came at the expense of an underused workspace server acting as the grid server client.

An alternative would be to have the SAS Object Spawner use the grid to launch a SAS Workspace Server so that the grid’s management and policy could be applied. A new load-balancing method is available to grid administrators in SAS Grid Manager 9.4 to launch the workspace servers using a logical grid server defined in the same SAS Application Server. Using this load-balancing method, a workspace server will be launched on the grid anywhere an object spawner is installed and configured as part of the SAS Application Server’s logical workspace server. Figure 5 shows the workspace server’s Load Balancing Options dialog box, including a new option for using the grid to launch the servers.

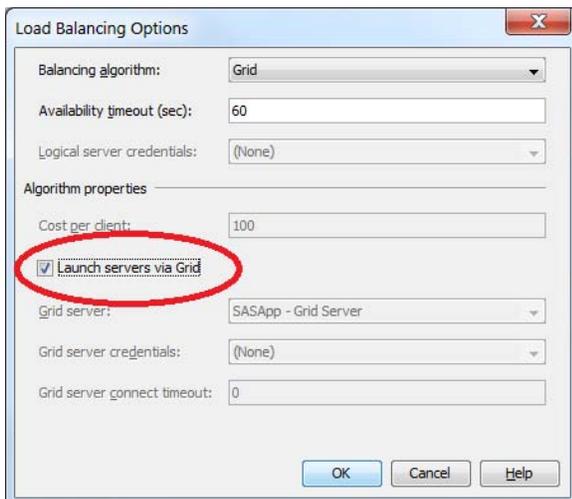


Figure 5. SAS Workspace Server Load Balancing Options

There are several advantages to using the grid to launch workspace servers beyond the ability to apply policy. They are as follows:

- elimination of GRIDWORK and RMTWORK libraries: There is only SASWORK.
- elimination of a grid server process: There is now only a SAS workspace server process.
- elimination of confusion when trying to monitor the user's SAS process: Now the same SAS process can be viewed from both SAS Management Console and Platform RTM for SAS.

In addition, applications that are able to tell the object spawner their names will be able to have the proper grid options set applied when starting the workspace server. For example, both SAS Enterprise Guide and SAS Data Integration Studio tell the object spawner their application names. When the object spawner launches a workspace server using the grid, it will check for a grid options set for the respective application and the current user, and will use the additional information if present. With this capability, you could have SAS Enterprise Guide users launching workspace servers by submitting a job to a different queue than SAS Data Integration Studio users.

SASGSUB ENHANCEMENTS

It is commonly the case that SAS customers use an enterprise scheduler that has been chosen by their IT organization as their standard scheduler. Common examples would include BMC Control-M and CA 7. In addition, they have considerable investment in their enterprise scheduler with the number of jobs that already use the scheduler, the existing knowledge of the scheduler by their users, as well as integration with incident management and accounting systems. It is also the case that when deploying SAS in a SAS grid environment, there is the need to be able to submit all of the SAS workload to the grid. However, it is typically cost prohibitive and impractical for such a customer to learn and convert to the scheduling capabilities that are part of the Platform Suite for SAS that is shipped with SAS Grid Manager.

To be able to use the grid with other workload automation before SAS 9.4, SAS recommended that you use SASGSUB in a script to do the following:

1. Run SASGSUB to submit a SAS program to the grid.
2. Run SASGSUB to determine the status of the grid job. Repeat this step until the grid job has completed.
3. Run SASGSUB to retrieve the results of the grid job.
4. Parse the results to determine whether the grid job succeeded or failed.

SAS even provides shell scripts to do this “submit/wait/get results” looping, but they are less than ideal. Starting in the second maintenance release of SAS 9.3, SASGSUB had a GRIDWAIT option to merge step #1 and step #2, but you still had to at least perform step #1 along with steps #3 and #4.

In the 9.4 version of the SASGSUB, a new option, GRIDWAITRESULTS, has been added so that SASGSUB will submit a SAS program, wait for the grid job to complete, return the results to the local directory, and set its return code to the return code from the grid job. After the results have been copied to the local directory, it will also copy the log and output files out of the results subdirectory into the current directory as if SAS was running in the current directory. With this new option, a script file to run SASGSUB is no longer needed. It can be replaced with a single call to SASGSUB and use its return code to determine whether this step in the workflow failed.

In addition, SAS Grid Manager 9.4 introduces the GRIDWAITTIMEOUT parameter so that you can specify the number of seconds SASGSUB should wait for the job to complete before returning an error.

ENHANCED DEBUGGING

The grid is a complex ecosystem with many moving parts, which makes it difficult at times to debug what is wrong. As an additional complication, the parts in SAS that make up the grid often have different methods of generating debugging information. Each puts their debugging information in separate files. Not only does this make things difficult to setup, but it also makes it difficult to match up the debugging information from various files.

In an effort to consolidate all information from the various SAS processes, SAS introduced the SAS logging facility in SAS 9.2. In SAS 9.4, many of the components that make up the grid have converted their debug output to use the SAS logging facility. Although the SAS Object Spawner and parts of the SASGSUB already used the SAS logging facility, the following subsystems have also incorporated its use for their entire debug and trace output.

The following loggers are now available in SAS 9.4:

- SAS/ CONNECT Server App.Connect.Server
- SAS/CONNECT Client App.Connect.Client
- SAS/CONNECT Spawner App.Connect.Spawner
- SAS TCP/IP subsystem App.WQ
- SAS Grid Manager Client Utility App.Grid
- SAS Grid Manager Grid Provider App.tk.tkegrid
- SAS/SECURE App.tk.eam

In addition, the SASGRID script that is installed on every grid node to start the grid server process has also added new logging support. The SASGRID script will always generate a log in the configured logs directory under the name "**SASGrid.<user>.<hostname>.<process_id>.<YYYYMMDDhhmmss>.log**" so that errors unrelated to SAS, but related to running SAS on the grid, are saved. If the SAS return code is 0 ('no errors'), 1 ('SAS processing warnings'), or 2 ('SAS processing errors'), this file will be automatically deleted. If any other error occurs, this file will remain so that the grid administrator can determine what went wrong.

CONCLUSION

By adding grid options sets, grid-launched workspace servers, new wait capabilities to SASGSUB, and enhanced logging through the SAS Logging Facility, SAS 9.4 brings about many new features to help customers more easily manage a SAS Grid Manager grid.

RECOMMENDED READING

- *Grid Computing in SAS® 9.4*
- *SAS® 9.4 Logging: Configuration and Programming Reference*

CONTACT INFORMATION

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

Doug Haigh
SAS Institute, Inc.
500 SAS Campus Drive
Cary, NC 27513
Doug.Haigh@sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.