

## Deep Dive with SAS® Studio into SAS® Grid Manager 9.4

Edoardo Riva, SAS Institute Inc.

### ABSTRACT

Do you know how many different ways SAS® Studio can run your programs with SAS® Grid Manager? SAS Studio is the latest and coolest interface to SAS® software. As such, we want to use it in most situations, including with sites that leverage SAS Grid Manager. Are you new to SAS and want to be guided by a modern UI? SAS Studio is here to help you. Are you a code fanatic, who wants total control of how your program runs to harness the full power of SAS Grid Manager? Sure, SAS Studio is for you, too. This paper covers all SAS Studio editions. You learn how to connect each of them to SAS Grid Manager and discover best practices for harnessing a high-performance SAS analytics environment, while avoiding potential pitfalls.

### INTRODUCTION

With SAS® Studio, you can access your data files, libraries, and existing programs, and you can write new programs. You can also use the predefined tasks in SAS Studio to generate SAS® code for you.

SAS® Grid Manager lets you balance workloads. It also gives you faster parallel processing, high availability, and enterprise scheduling, all in a flexible and centrally managed grid computing environment.

They seem such different technologies, yet they can provide great benefits when used together. This paper concentrates on three grid features, showing how SAS Studio can help you take advantage of workload balancing and parallel computing, while using grid central-management capabilities.

### SAS STUDIO

SAS Studio is a multi-functional application for leveraging the power of SAS through your web browser.

When you run a program or task, SAS Studio connects to a SAS server to process the SAS code. The SAS server can be a hosted server in a cloud environment, a server in your local environment, or a copy of SAS on your local machine. For server-based environments, including SAS Grid Manager provides the extra capabilities discussed in this paper.

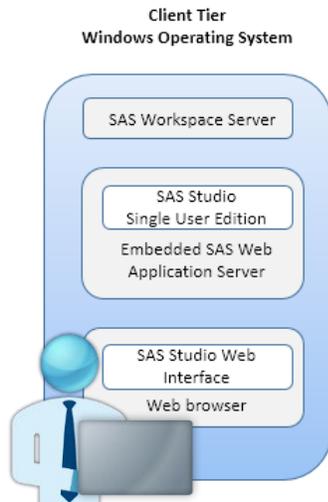


After the code is processed, the results are returned to SAS Studio in your browser.

### SAS STUDIO EDITIONS

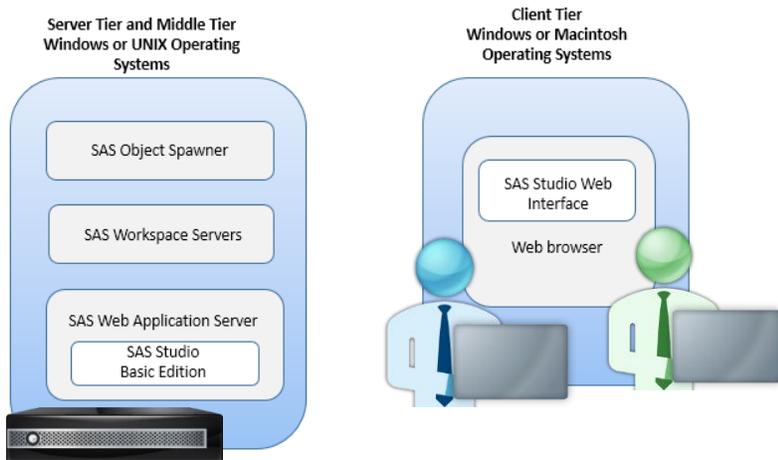
SAS Studio is available in three deployments: SAS Studio Single-User, SAS Studio Basic and SAS Studio Enterprise Edition.

The single-user edition of SAS Studio is delivered with every copy of Base SAS® and runs on Windows operating environments. All the software components of SAS Studio are installed on the same machine, and only one user identity is allowed access. You can think of it as the web version of the traditional SAS Display Manager System.



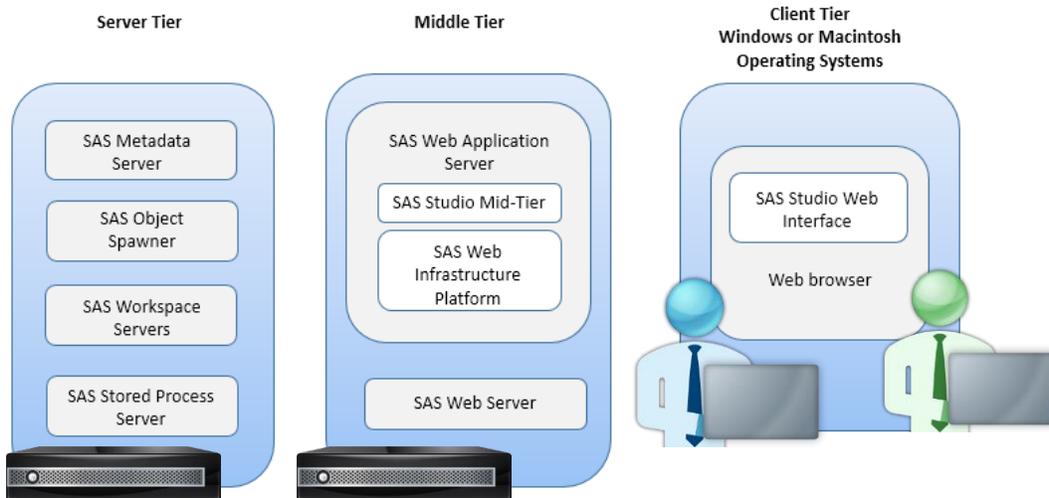
**Figure 1. SAS Studio Single-User Edition High-Level Architecture**

The basic edition of SAS Studio is delivered with Base SAS and runs on Windows and UNIX operating environments. This edition includes the SAS® Web Application Server and the SAS® Object Spawner. Any user who has an operating system account on the Windows or UNIX server machine can log on through a web browser over the network.



**Figure 2. SAS Studio Basic Edition, High-Level Architecture**

The enterprise edition of SAS Studio is available with the SAS® Integration Technologies license, which is included in most of SAS solutions. This edition includes the SAS® Metadata Server, the SAS Web Application Server, the SAS® Web Server, and the SAS® Web Infrastructure Platform services, applications, and data server.



**Figure 3. SAS Studio Enterprise Edition, High-Level Architecture**

Why bother with all these versions? Because all editions of SAS Studio can take advantage of the processing capabilities of a SAS® Grid, but the approach to take depends on the edition of SAS Studio that you are using. All this is described in detail in the paper.

Note: SAS® University Edition includes SAS Studio together with Base SAS, SAS/STAT® software, SAS/IML® software, SAS/ACCESS® software, and several time series forecasting procedures from SAS/ETS® software. This paper does not include a discussion of SAS University Edition because it does not have the components required to connect to a grid environment.

## SAS STUDIO RELEASES

SAS Studio was initially released in 2014 with the first maintenance release of SAS® 9.4. It has been updated several times since then, as documented in Table 1.

| SAS Studio Release | Supported SAS Release                           |
|--------------------|---|
| SAS Studio 3.1     | SAS 9.4 TS1M1, ship event 14W11 (March 2014)    |
| SAS Studio 3.2     | SAS 9.4 TS1M2, ship event 14W32 (August 2014)   |
| SAS Studio 3.3     | SAS 9.4 TS1M2, ship event 15W08 (February 2015) |
| SAS Studio 3.4     | SAS 9.4 TS1M3, ship event 15W29 (July 2015)     |
| SAS Studio 3.5     | SAS 9.4 TS1M3, ship event 16W08 (February 2016) |

**Table 1. SAS Studio Releases**

Many new features have been added since it was initially released. If you want to be able to harness all its power, be sure to be running the most current release. Otherwise, it is time for an upgrade!

## SAS GRID MANAGER

SAS Grid Manager provides a modern, flexible infrastructure that turbo-charges SAS performance with distributed and parallel computing techniques. All under the automatic monitoring, resource management, and orchestration of a grid controller. When you leverage a SAS grid computing environment, you can automatically distribute SAS computing tasks among multiple computers on a network under the control of SAS Grid Manager.

Starting with the third maintenance release of SAS 9.4, released in, July 2015, when you license SAS Grid Manager you can choose one of the two available flavors. SAS Studio works seamlessly with either edition.

## SAS GRID MANAGER WITH PLATFORM SUITE FOR SAS

Platform Suite for SAS is a set of components, provided by IBM Platform Computing, that provide efficient resource allocation, policy management, and load balancing of SAS workload requests.

SAS Grid Manager with Platform Suite for SAS includes all of the required grid software components, both from Platform Computing and SAS.

## SAS® GRID MANAGER FOR HADOOP

SAS Grid Manager for Hadoop provides workload management, accelerated processing, and scheduling of SAS analytics co-located on a Hadoop cluster. SAS Grid Manager for Hadoop leverages YARN to manage resources and distribute SAS analytics to a Hadoop cluster running multiple applications. It integrates with Oozie, which provides scheduling capability for SAS workflows. SAS Grid Manager for Hadoop supports all of the existing SAS Grid syntax, submission modes, and integration with other SAS products and solutions.

SAS Grid Manager for Hadoop does not include a Hadoop distribution or any of the Hadoop components. However, you must have one of the supported enterprise Hadoop distributions already installed and configured.

## HOW DOES SAS STUDIO LEVERAGE A GRID ENVIRONMENT?

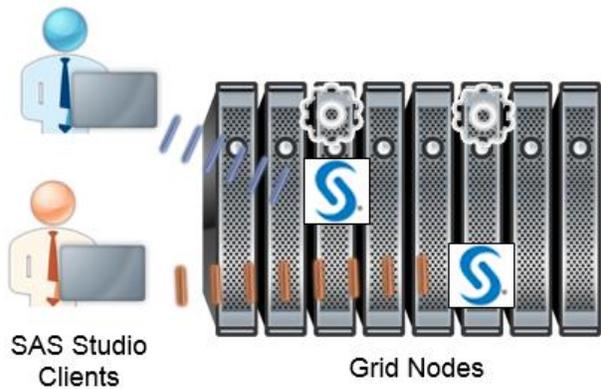
As you can see in the figures in the previous sections, every edition of SAS Studio uses workspace server sessions to run SAS code. In order to be able to run it on a grid, you must have SAS Grid Manager licensed on the same machines where the SAS Studio workspace servers are started. These will be your grid entry point.

With all editions of SAS Studio, you can use in your code any of the functions included with the SAS Grid Manager license, such as `GRDSVC_ENABLE`, to send the code to the grid for remote execution. In this way, you can start multiple sessions in parallel to run your code faster and get results quickly. For example, Figure 4 shows a SAS user running three parallel grid sessions, all started by one of the workspace servers used by SAS Studio.



**Figure 4. Parallel Code Execution Leveraging SAS/CONNECT®**

If you are using SAS Studio Enterprise Edition together with SAS Grid Manager, your administrator can configure the SAS Studio workspace servers to use load balancing and then select to have the grid launch the workspace servers. After that, without any code change or end-user intervention, the grid control server automatically starts new server sessions on the best available grid node, as defined by the current load on the grid hosts and the policies set by the administrator. Figure 5 shows two workspace server sessions, each servicing a different end user, automatically load-balanced by SAS Grid Manager and started on different grid nodes.

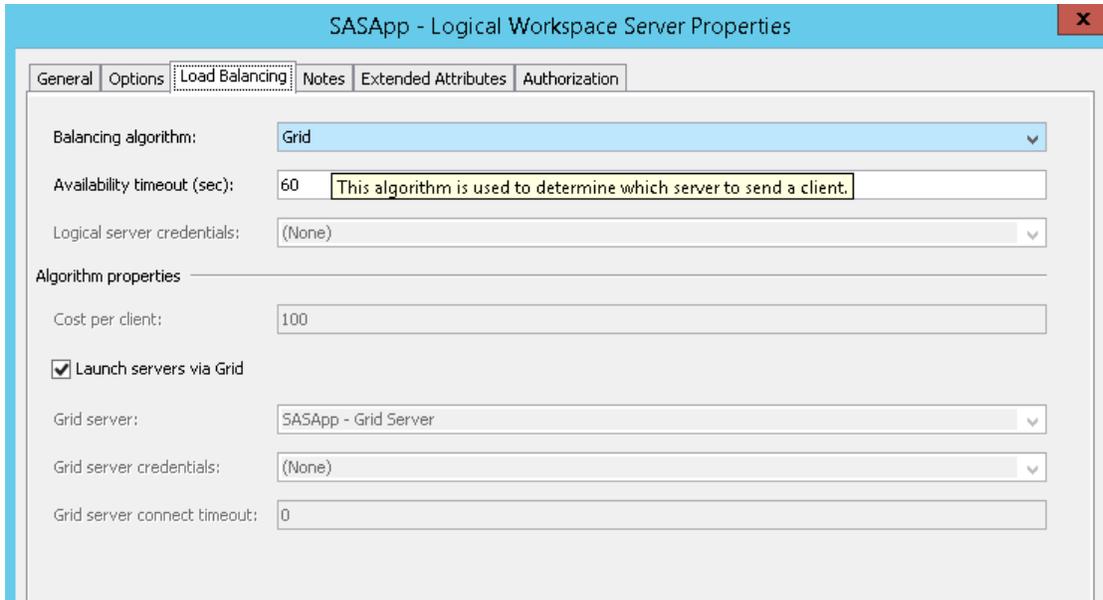


**Figure 5. Multi-User Workload Balancing with Grid-Launched Workspace Servers**

Let's see both use cases in more detail. We will actually start with the second one; it is simply so amazing that you will find yourself using the grid without even knowing it!

### MULTI-USER LOAD BALANCING

SAS® Workspace Servers are capable of performing load balancing across multiple machines. These servers can be configured to use one of the default algorithms to provide load balancing. However, with SAS Grid Manager installed, an administrator can configure workspace servers and other SAS servers to use SAS Grid Manager to provide load balancing. This is a one-time configuration done using SAS® Management Console, shown in Display 1. Any SAS product or solution that uses workspace servers, including SAS Studio Enterprise Edition, will benefit from using SAS Grid Manager to provide load balancing. However, using the grid to provide load balancing also increases overhead, so each session might take a few seconds longer to start.

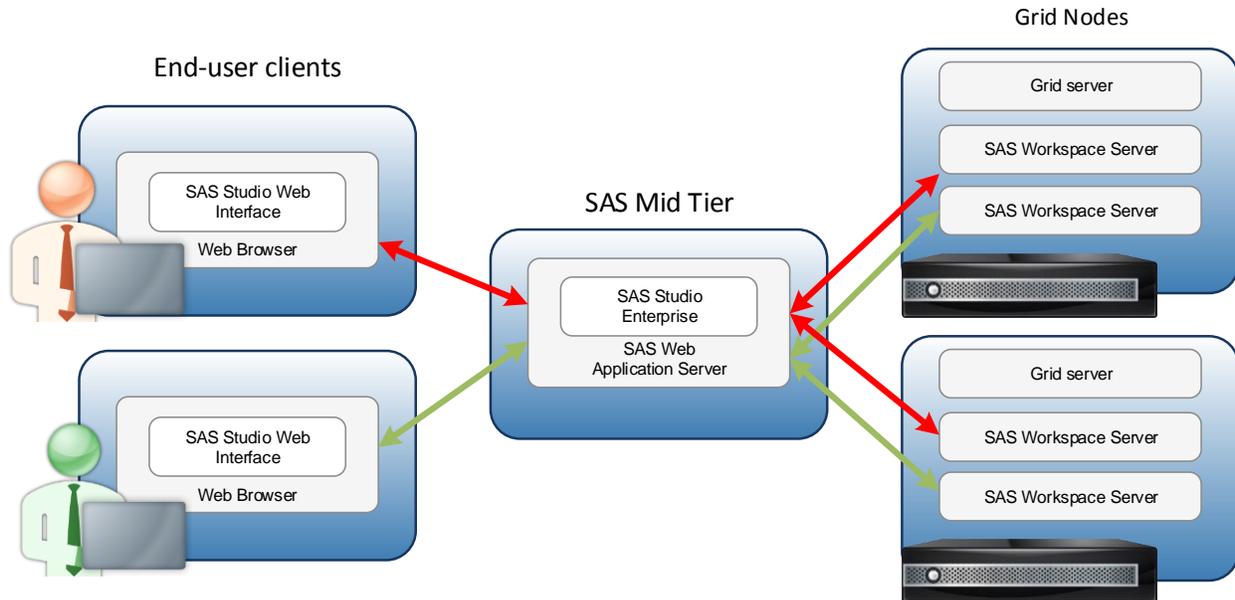


**Display 1. Server Load-Balancing Options Dialog Box in SAS Management Console**

For you, as the end user, SAS Studio behavior is unchanged. Simply sign on and, even before pressing a button or writing any code, you get two workspace server sessions running on your grid hosts. On a side note, SAS Studio always starts at least two sessions. One is used to run the user code, while the other is used for file I/O and other internal operations.

As soon as another user signs on, two additional sessions appear on the grid, and so on for each additional user. The grid controller takes care of starting each session on the best available server, resulting in evenly spread resource utilization.

Figure 6 shows two end users that simply signed on to SAS Studio Enterprise Edition. SAS Grid Manager took care of starting their sessions load balancing them on the grid nodes.



**Figure 6. Workspace Servers Load Balanced across Grid Nodes**

So far, we have only opened the SAS Studio web interface. What happens when we start using it? Remember that every code that is generated by SAS Studio or every action that requires querying the back-end SAS session executes in one of the workspace servers. This means that everything that runs is automatically using the grid.

## REMOTE CONNECT TO GRID SESSIONS

With all editions of SAS Studio, you can use SAS/CONNECT statements in your code to send the code to the grid. It is really simple to convert every program so that it runs on a grid—all you do is add five extra lines of code:

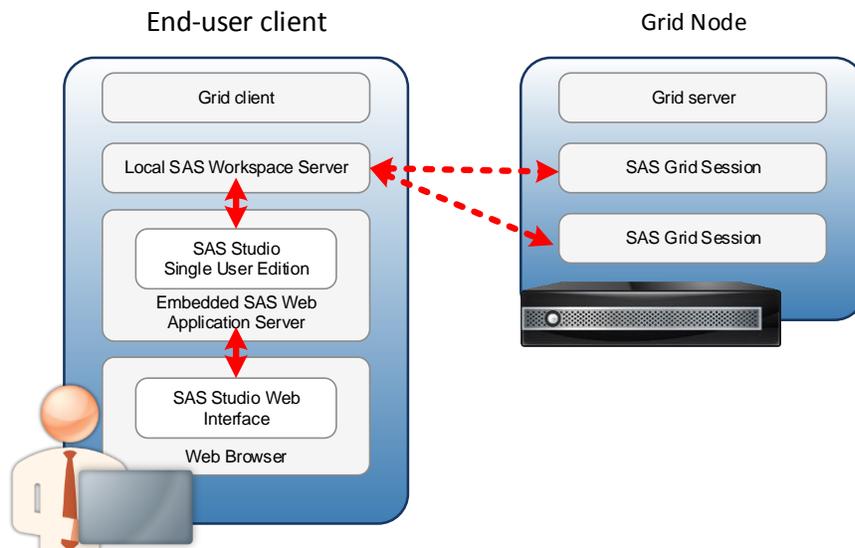
```
GRIDSVC_ENABLE
SIGNON
RSUBMIT
ENDRSUBMIT
SIGNOFF
```

These statements are not described in detail in this paper. You can find a very good explanation in Doug Haig's paper from SAS Global Forum 2015 listed in the references section.

The result is that the workspace server session, which is currently running your SAS Studio code, launches one or more additional remote grid sessions. All the code between the RBSUBMIT/ENDRSUBMIT blocks is forwarded there for execution on the grid. Some SAS programs contain multiple independent subtasks that can be executed in parallel. Just add RSUBMIT and ENDRSUBMIT statements around each subtask and SAS Grid Manager automatically assigns each identified subtask to a grid node.

Figure 7 shows an example of SAS Studio single-user edition in which the end user has submitted the statements to launch two remote parallel grid sessions. For simplicity, only one of the two workspace server sessions that SAS Studio always uses is shown. The dotted lines represent the connections instantiated by the SAS/CONNECT protocol. Also, note that since the workspace server session is acting

as a grid client, the client components of the grid software are required to be installed and configured on the end-user client. These client components are the Platform Load Sharing Facility (LSF) client for the SAS Grid Manager for Platform, or Hadoop client JAR files and XML configuration files for SAS Grid Manager for Hadoop.



**Figure 7. SAS Studio Single-User Edition Remotely Connected to a Grid**

## DIFFERENT MODES OF EXECUTION INSIDE SAS STUDIO

SAS Studio wants to be your interface of choice, whatever your habits and your programming style are. That is why it includes two different *perspectives*: the SAS Programmer perspective and the Visual Programmer perspective. It also supports different code submission modes: noninteractive, interactive, and batch. You can find all the details of what these are and how to use them in the official documentation, but it is interesting to explore here how some of these interact with a grid environment.

### BATCH SUBMIT

Starting with SAS Studio 3.5, you can submit SAS programs in batch. As you can imagine, this lets you submit a program, close your browser and then come back later to check the results.

If you are familiar with SAS architecture, it will be interesting to know that this batch submission does not use the SAS® DATA Step Batch Server. Just as with the other code submission modes, SAS Studio starts a workspace server; the difference here is that it keeps the server running on your behalf even after you log off from the web interface. This is useful to know while configuring or monitoring the grid back end: from the grid point of view, these batch sessions are exactly the same as the interactive ones!

One interesting aspect of batch SAS Studio submissions is documented in the product Administrator's Guide. The guide explains how to set some properties in the SAS Studio configuration file to limit the maximum number of active batch sessions per user or across all users. Table 2 lists these properties with their default value.

| Property                            | Description   |
|-------------------------------------|---|
| webdms.maxNumActiveBatchSubmissions | <p>Specifies the maximum number of active batch jobs for the current SAS Studio user. The default value depends on your edition of SAS Studio.</p> <p>For the SAS Studio Enterprise Edition and SAS Basic Edition, the default value is <b>3</b>.</p> <p>For the SAS Studio Single-User Edition, the default value is <b>5</b>.</p> |

|   |  |
|---|--|
| webdms.maxNumActiveBatchSubmissionsSystem | <p>Specifies the maximum number of batch jobs that can be submitted for a given instance of SAS Studio across all users. The default value depends on your edition of SAS Studio.</p> <p>For the SAS Studio Mid-Tier (Enterprise) Edition and SAS Basic Edition, the default value is 24.</p> <p>For the SAS Studio Single-User Edition, the default value is 5.</p> |
|---|--|

**Table 2. SAS Studio Properties to Limit the Number of Concurrent Batch Submissions**

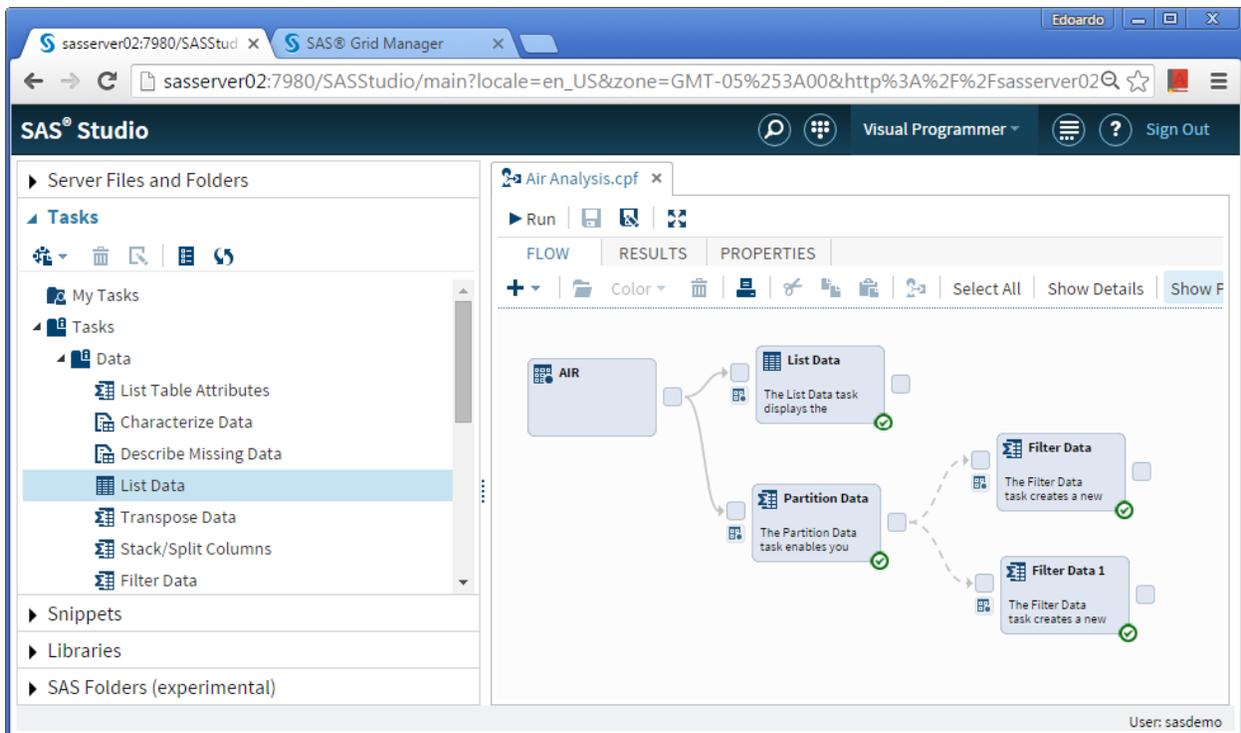
In your grid environment, these default values might be too small, especially if you have many cores. You can ask your administrator to increase them if you run out of available sessions.

## PARALLEL PROCESS FLOWS

One of the hidden gems of SAS Studio that really shines when used in a grid environment is the capability of running process flows in parallel. However, let's do one step at a time: what are process flows?

When working in the Visual Programmer perspective, you have access to process flows. A process flow is a graphical representation of a process, where each object, be it a SAS program, a SAS Studio task, a query, and so on, is represented by a node. Nodes are connected by links that instruct SAS Studio how to move from one node to the next one.

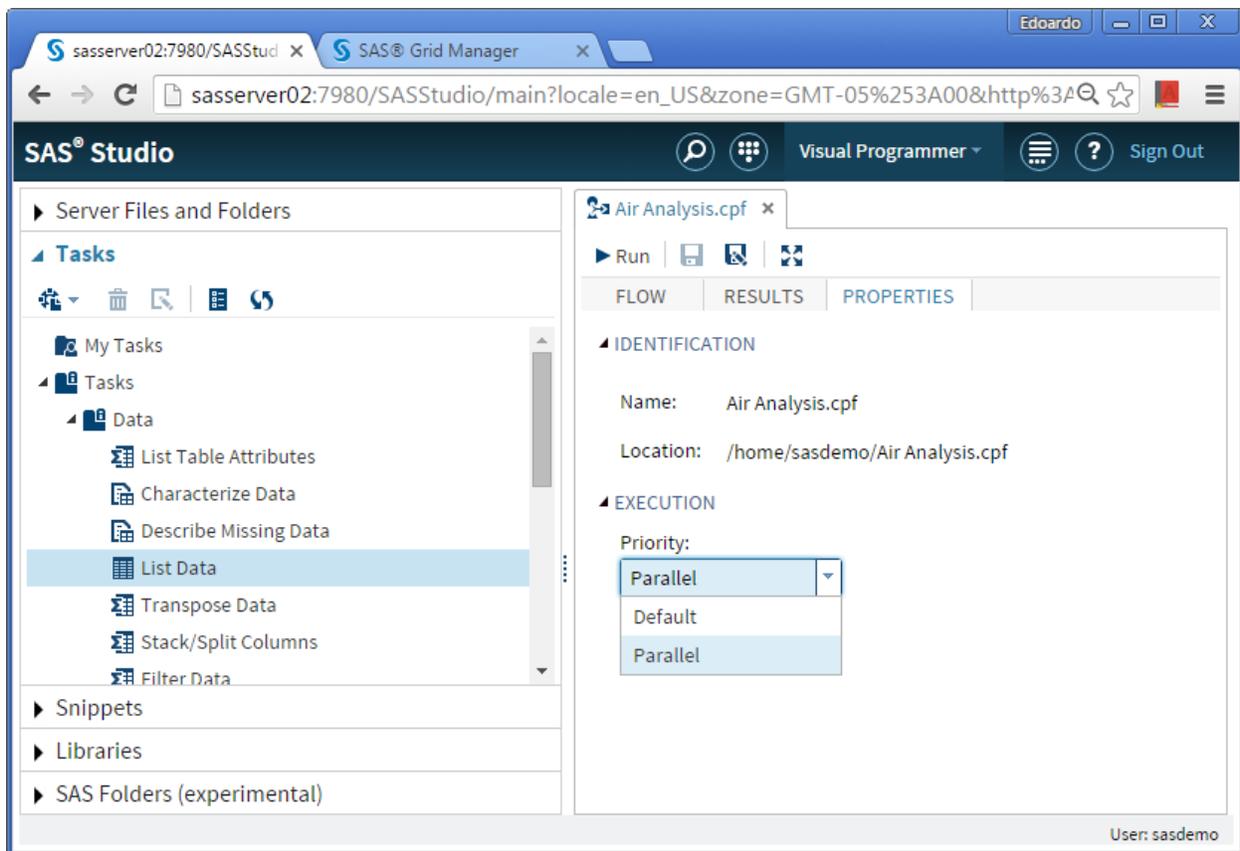
Display 2 shows a simple SAS Studio process flow.



**Display 2. SAS Studio Process Flow**

On the **Properties** tab of the current process flow, you can set the execution mode of the nodes. With the default setting, SAS Studio runs the nodes in the order in which they are added to the process flow. If node 2 is dependent on another node 1, node 1 must run completely before node 2 will run.

You can change the execution mode to **Parallel** as shown in Display 3. When this value is set, SAS Studio uses multiple workspace servers to run the nodes concurrently, always enforcing the correct dependencies.



### Display 3. Setting the Execution Mode to Parallel

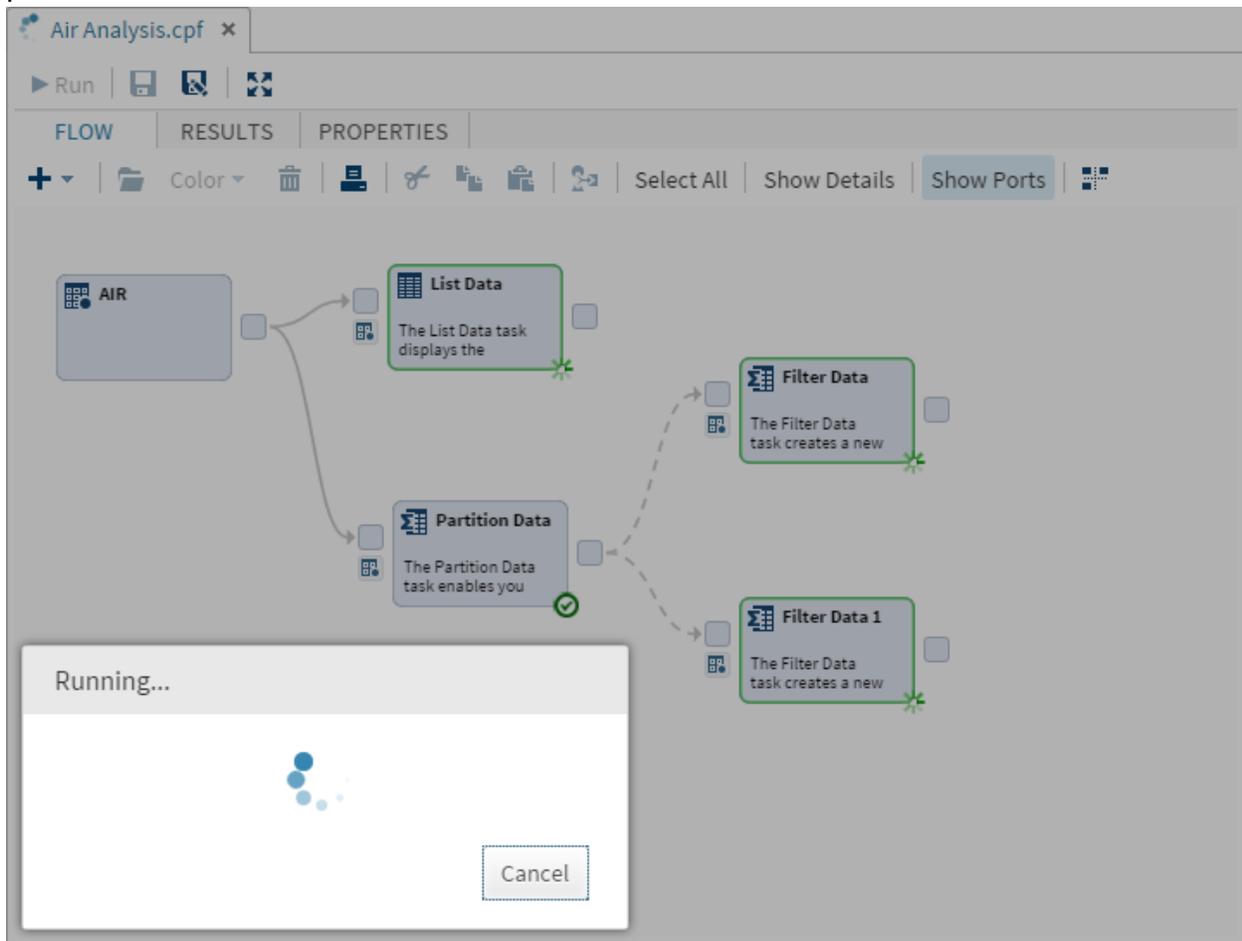
When you use this feature in a grid environment, you can achieve the benefits and the performance improvements of multi-machine parallel load balancing, without having to code any SAS/CONNECT statement. It's a real point-and-click parallel execution engine!

Display 4 shows the process flow presented in Display 2 while it is running in parallel execution mode. The pane is grayed out because it is not possible to interact with it until the execution is complete.

We can see that the List Data node is still running, while the Partition Data node has already finished. Thus, the two Filter Data nodes were able to start in parallel.

In this scenario, we would guess three workspace server sessions are concurrently running our code in the grid. However, if we monitor what is happening on the back-end hosts, we notice something unexpected. There are actually five workspace server sessions running. Why? If you remember, when you sign in to SAS Studio, it starts two SAS sessions. These are used only for the default execution mode. If a process flow is run in parallel mode, up to three additional SAS session are started, for a total of five. Once the process flow is finished, the three additional SAS processes terminate if there is no further activity for 30 seconds, in order to release resources.

Just as with batch processing, an administrator can use a configuration property, `webdms.maxParallelWorkspaces`, to specify the maximum number of workspaces that can be used when SAS is running in parallel mode. The default value is 3. The maximum value is 8.



**Display 4. Tasks Running in Parallel in a Process Flow**

## PERSISTENCE OF SESSIONS

Parallel processing can speed up your projects by an incredible factor, especially when programs consist of subtasks that are independent units of work and can be distributed across a grid and executed in parallel. However, when these parallel execution environments are not kept in sync, it can also introduce unforeseen problems. The most common is the “disappearing” of temporary tables. This can actually happen using different client interfaces because this problem does not depend on using a certain software, but rather on the business logic that is implemented. Let’s discuss this problem with a practical example when using SAS Studio.

## WORK AND OTHER LIBRARIES

In this example, we want to run an analysis—here a simple PROC PRINT—on two independent subsets of the same table. We decide to use two parallel grid sessions to partition the data, and then we run the analysis in the parent session. The code we submit in SAS Studio could be similar to the following:

```
%let rc = %sysfunc( grdsvc_enable(_all_, server= SASApp));

signon grid1;
signon grid2;
```

```

proc datasets library=work noprint;
  delete sedan SUV;
run;

rsubmit grid1 wait=no ;
data sedan;
  set sashelp.cars;
  where Type="Sedan";
run;
endrsubmit;

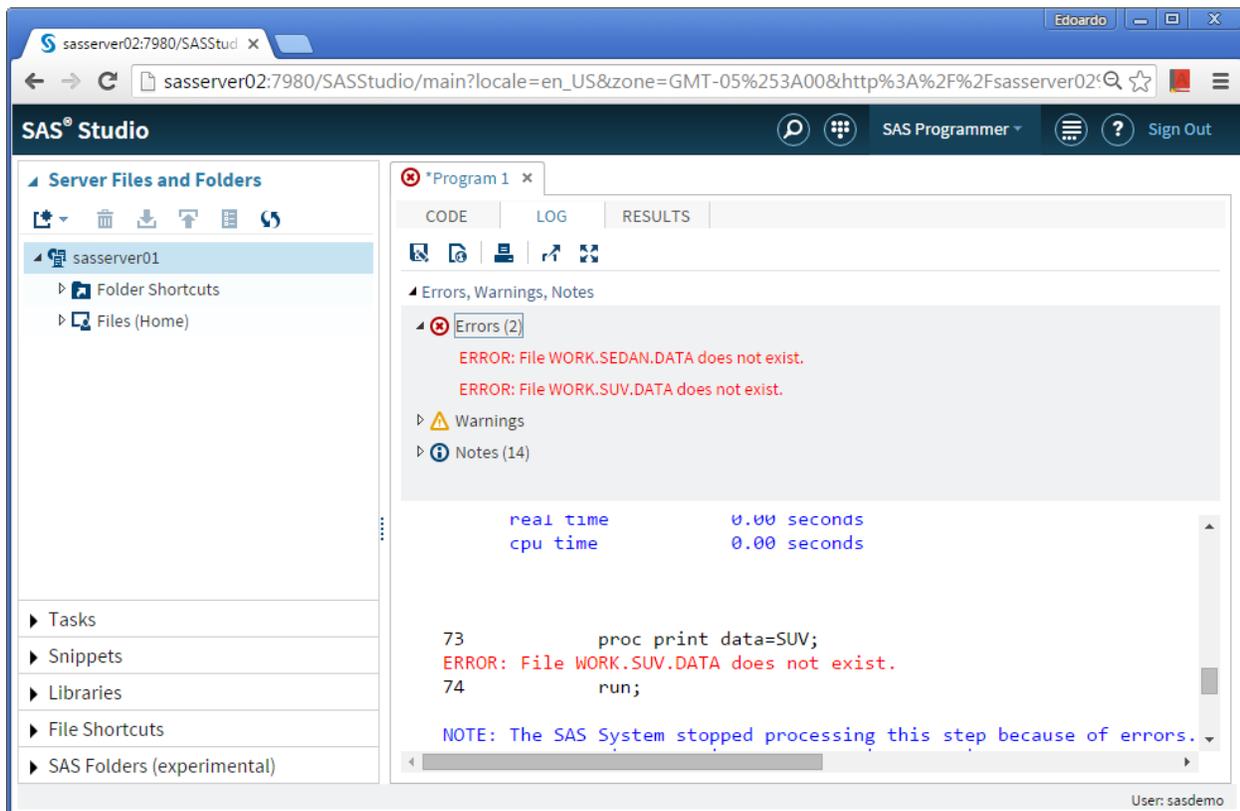
rsubmit grid2 wait=no ;
data SUV;
  set sashelp.cars;
  where Type="SUV";
run;
endrsubmit;

waitfor _ALL_ grid1 grid2;

proc print data=sedan;
run;
proc print data=SUV;
run;

```

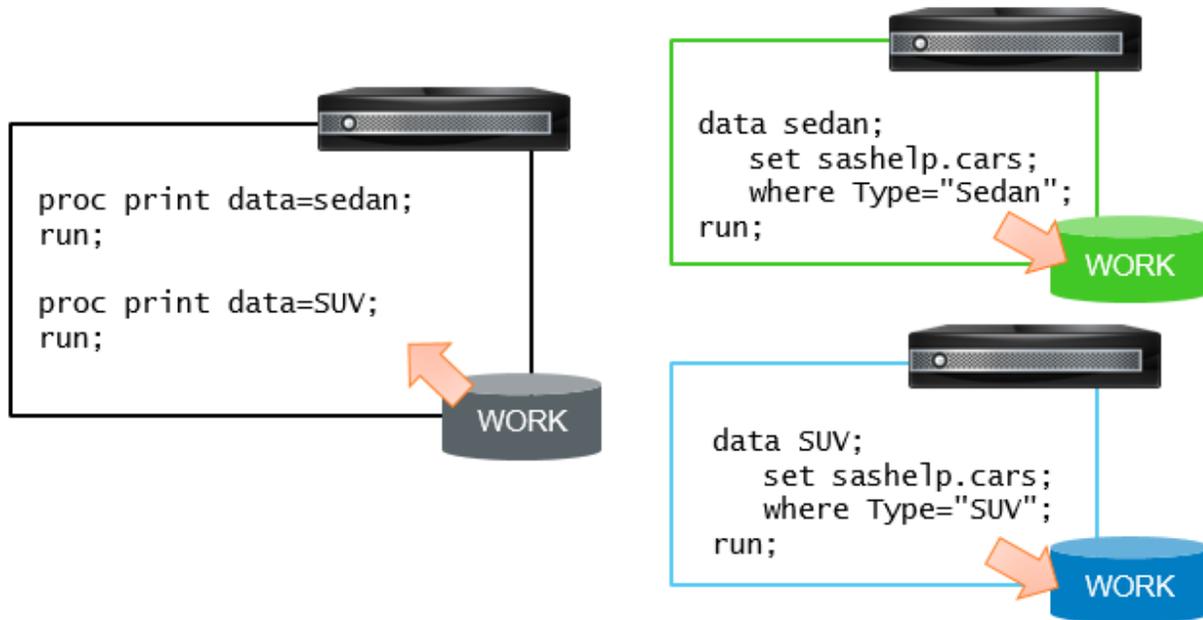
After submitting the code by pressing F3 or clicking **Run**, we do not get the expected Result window and the Log window shows some errors, as shown in Display 5.



Display 5. SAS Studio Log Window with Errors

We get these results because we are using the WORK library, the temporary library that is automatically defined by SAS at the beginning of each SAS session or job. The WORK library stores temporary SAS files that are written by a DATA step or a procedure and then read as input of subsequent steps. When we request parallel execution on the grid, tasks run in multiple SAS sessions, and each grid session has its own dedicated WORK library that is shared neither with any other grid session, nor with the parent session that started it.

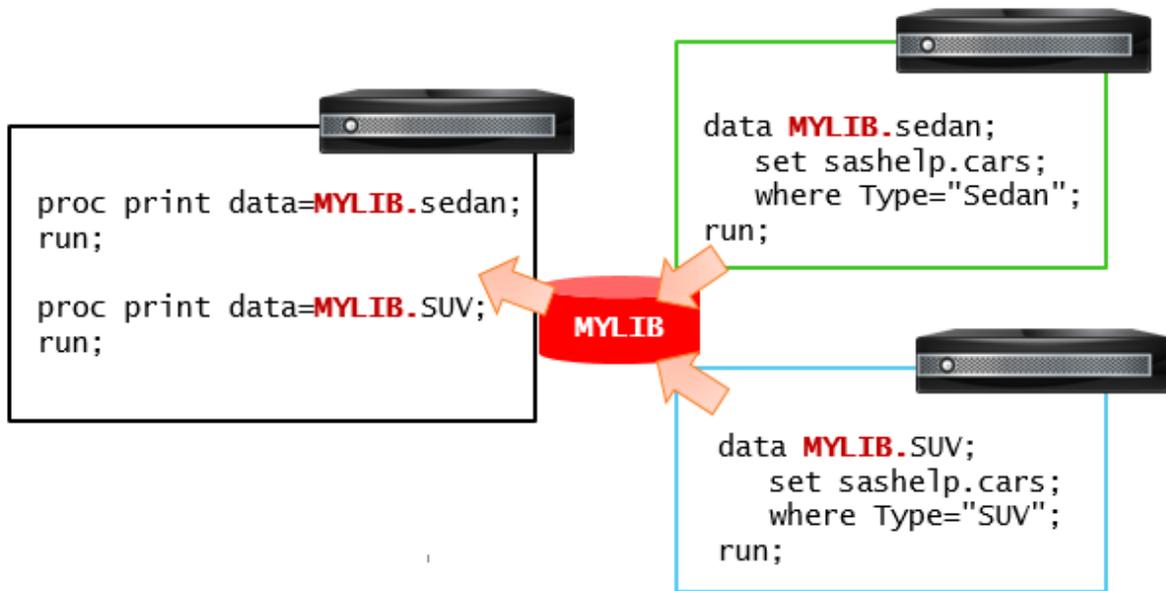
In the above example, the DATA steps output their results—the SEDAN and SUV tables—in the WORK library of a SAS session. Then, PROC PRINT tries to read those tables from the WORK library of a different SAS session. Figure 8 shows that the desired tables are not where we expect them, and the task fails.



**Figure 8. Incorrect Use of the WORK Library between Multiple Sessions**

This issue is quite common when dealing with multiple sessions, even without a grid. One simple solution is to avoid using the WORK library and any other non-shared resources. It is possible to assign a common library in many ways, such as in autoexec files or in metadata.

Figure 9 shows how a common library solves the issue.



**Figure 9. Correct Use of a Shared Library**

When submitting process flows in the Visual Programmer perspective in parallel, SAS Studio can help us avoid this problem. We can save our intermediate results in the WEBWORK library. This special library is automatically assigned at start-up and is shared across all workspace server sessions.

As you might have guessed, libraries are not the only objects that should be shared across sessions. Every local setting—be it the value of an option, a macro, or a format—has to be shared across all parallel sessions. It is not difficult, but we have to remember to do it!

### WHERE ARE MY PREFERENCES?

Even when we use all due diligence in coding and using shared locations for all SAS artifacts, we can still fall in some sharing issues when using SAS Studio in a grid environment.

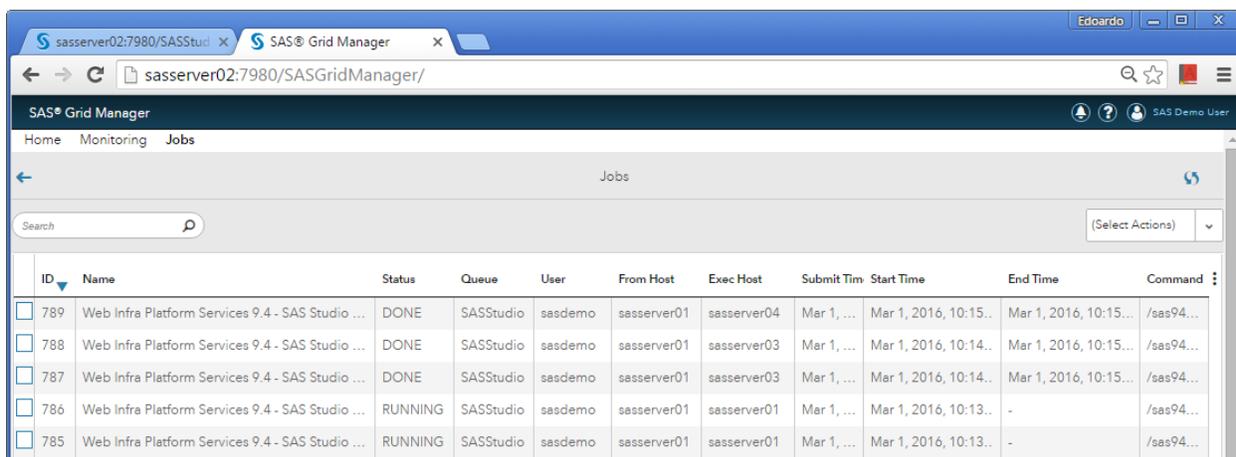
SAS Studio has a Preferences window that enables you to customize several options that change the behavior of different features of the software. By default, these preferences are stored under the end-user home directory on the server where the workspace server session is running (%AppData%/SAS/SASStudio/preferences in Windows or ~/.sasstudio/preferences in UNIX). Does this sentence ring any alarm bell? With SAS Studio Enterprise Edition running in a grid environment, there is no such thing as “the server where the workspace server session is running”! One time it runs on one grid node, a few minutes later it runs on another one. For this reason, it might happen that a preference that we just set to a custom value reverts to its default value on the next sign-in. This issue can become worse because SAS Studio follows the same approach to store code snippets, tasks, autosave files, and the WEBWORK library.

Until SAS Studio 3.4, the only solution to this uncertainty was to have end users’ home directories shared across all the grid nodes. SAS Studio 3.5 removes this requirement by providing the administrators with a new configuration option, `webdms.studioDataParentDirectory`. This option specifies the location of SAS Studio preferences, snippets, my tasks, and more. The default value is blank, which means that the behavior is the same as in previous releases. An administrator can point it to any shared location to access all of this common data from any workspace server session.

## MONITORING SAS STUDIO SESSIONS

If your grid uses Platform Suite for SAS, SAS® Environment Manager provides a management module that enables administrators and end users to monitor a SAS grid cluster. While a grid administrator can see and manage the whole grid including jobs submitted by any user, end users can take advantage of this module to monitor their own sessions. When using SAS Grid Manager for Hadoop, a similar monitoring is provided by Hadoop using the YARN Resource Manager Web User Interface.

Display 6 displays an end user monitoring the SAS processes running on the grid after submitting the process flow shown in Display 2 and Display 4. We can recognize the two initial sessions, labeled ID 785 and 786 and Status of RUNNING. The three additional sessions used for parallel execution have ID 787, 788 and 789 and have a status of DONE because SAS Studio has already terminated them, after 30 seconds of inactivity.



The screenshot shows the SAS Grid Manager web interface. The browser address bar indicates the URL is `sasserver02:7980/SASGridManager/`. The page title is "SAS® Grid Manager" and the user is "SAS Demo User". The navigation menu includes "Home", "Monitoring", and "Jobs". The "Jobs" section is active, displaying a table of job execution details. The table has columns for ID, Name, Status, Queue, User, From Host, Exec Host, Submit Time, Start Time, End Time, and Command. There are five rows of data, with IDs 785, 786, 787, 788, and 789. Jobs 785 and 786 are in a "RUNNING" state, while jobs 787, 788, and 789 are in a "DONE" state.

| ID  | Name   | Status  | Queue     | User    | From Host   | Exec Host   | Submit Time | Start Time           | End Time              | Command   |
|-----|--|---------|-----------|---------|-------------|-------------|-------------|----------------------|-----------------------|-----------|
| 789 | Web Infra Platform Services 9.4 - SAS Studio ... | DONE    | SASStudio | sasdemo | sasserver01 | sasserver04 | Mar 1, ...  | Mar 1, 2016, 10:15.. | Mar 1, 2016, 10:15... | /sas94... |
| 788 | Web Infra Platform Services 9.4 - SAS Studio ... | DONE    | SASStudio | sasdemo | sasserver01 | sasserver03 | Mar 1, ...  | Mar 1, 2016, 10:14.. | Mar 1, 2016, 10:15... | /sas94... |
| 787 | Web Infra Platform Services 9.4 - SAS Studio ... | DONE    | SASStudio | sasdemo | sasserver01 | sasserver03 | Mar 1, ...  | Mar 1, 2016, 10:14.. | Mar 1, 2016, 10:15... | /sas94... |
| 786 | Web Infra Platform Services 9.4 - SAS Studio ... | RUNNING | SASStudio | sasdemo | sasserver01 | sasserver01 | Mar 1, ...  | Mar 1, 2016, 10:13.. | -                     | /sas94... |
| 785 | Web Infra Platform Services 9.4 - SAS Studio ... | RUNNING | SASStudio | sasdemo | sasserver01 | sasserver01 | Mar 1, ...  | Mar 1, 2016, 10:13.. | -                     | /sas94... |

Display 6. Monitoring Job Execution Using the SAS Environment Manager Grid Module

## CONCLUSION

In this paper, you have seen how SAS Studio can leverage a grid environment, thanks to multi-user load balancing and remote connect to grid sessions. These capabilities empower all SAS Studio users to benefit from distributed and parallel computing techniques, under the automatic monitoring, resource management, and orchestration of a grid controller.

## REFERENCES

Haigh, Doug. 2015. "Divide and Conquer—Writing Parallel SAS® Code to Speed Up Your SAS Program." *Proceedings of the SAS Global Forum 2015 Conference*. Cary, NC: SAS Institute Inc.

Available at <http://support.sas.com/resources/papers/proceedings15/SAS1935-2015.pdf>.

## RECOMMENDED READING

- *Grid Computing in SAS® 9.4*.  
Available at <http://support.sas.com/documentation/onlinedoc/gridmgr/index.html>.
- *SAS Studio: User's Guide*.  
Available at <http://support.sas.com/documentation/onlinedoc/sasstudio/index.html>.
- *SAS Studio: Administrator's Guide*.

Available at <http://support.sas.com/documentation/onlinedoc/sasstudio/index.html>.

- *SAS Grid Computing Overview*

Available at <https://www.youtube.com/watch?v=BIK8JzDsSQg>.

- *Working in SAS Studio*

Available at

[https://www.youtube.com/watch?v=usnucvpnGLM&list=PLVBcK\\_IpFVi9cajJtRel2uBLbtCLz-WIN&index=4](https://www.youtube.com/watch?v=usnucvpnGLM&list=PLVBcK_IpFVi9cajJtRel2uBLbtCLz-WIN&index=4).

- *Riva, Edoardo. "SAS Studio on Grid."*

Available at <https://www.youtube.com/watch?v=ax-AbgjZs2M>.

- *Riva, Edoardo. "Avoid the pitfalls of parallel jobs."*

Available at <http://blogs.sas.com/content/sqf/2016/03/01/avoid-the-pitfalls-of-parallel-jobs/>.

## CONTACT INFORMATION

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

Edoardo Riva

SAS Institute

+1 919 531 7293

[edoardo.riva@sas.com](mailto:edoardo.riva@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.