Grid Computing in SAS® 9.4, Fifth Edition

SAS® Documentation
July 19, 2023

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *Grid Computing in SAS® 9.4, Fifth Edition*. Cary, NC: SAS Institute Inc.

Grid Computing in SAS® 9.4, Fifth Edition

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

July 2023

9.4-P5:gridref

Contents

PART 3 Grid Computing for SAS Using SAS Grid Manager for Platform

PART 4 Grid Computing for SAS Using SAS Grid Manager for Hadoop

PART 5 Common SAS Grid Manager Features

PART 6 SAS Grid Language Reference

PART 7 Appendix

What's New in SAS Grid Manager 9.4

Overview

SAS Grid Manager has the following new features and enhancements:

- In the SAS 9.4M7 security update, Log4J version 1 is replaced by Reload4j.
- In SAS 9.4M8, Log4J2.default release 2.17.1 is used.
- In SAS 9.4M8, the nmap command is used instead of the telnet command.
- In SAS 9.4M8, new fields are available when specifying general settings for a SAS Workload Orchestrator configuration.
- In SAS 9.4M6, a new grid provider has been added. It consists of SAS Workload Orchestrator and SAS Job Flow Scheduler. The product name is SAS Grid Manager.
- In SAS 9.4M6, the product name for a grid that uses Platform Suite for SAS has been changed to SAS Grid Manager for Platform.
- Support for grid options sets has been added.
- Support for grid-launched workspace servers has been added.
- New options have been added to the SAS Grid Manager Client Utility, including support for grid options sets, support for interactive SAS sessions and command processing, support for waiting for results from grid jobs, and support for submitting a SAS program located on a grid.
- New job options for Platform Suite for SAS have been added.
- SAS Grid Manager for Hadoop adds support for SAS grid processing co-located on a Hadoop cluster.
- In SAS 9.4M3, a SAS Grid Manager agent plug-in for SAS Environment Manager was added to support SAS Grid Manager for Platform. In SAS 9.4M5, metric data for grid queues was added to the SAS Grid Manager agent plug-in for SAS Environment Manager. In SAS 9.4M6, SAS Grid Manager agent plug-ins for SAS Environment Manager were added to support SAS Grid Manager and SAS Grid Manager for Platform.
- In SAS 9.4M3, a SAS Grid Manager module was added to SAS Environment Manager for SAS Grid Manager for Platform. In SAS 9.4M5, functions were added to specify dependencies and affinity requirements for high availability

(HA) configurations, to update the LSF license, and to terminate jobs, requeue jobs, change the position of jobs within a queue, and switch an uncompleted job to a different queue. In 9.4M6, the module's interface has been completely redesigned, although the functions provided remained the same.

- The GRDSVC_HOSTLIST and GRDSVC_OPTSETS functions have been added.
- In SAS 9.4M5, shortened option names are supported for the SAS Grid Manager Client Utility.
- In SAS 9.4M7, options have been added to the SAS Grid Manager Client Utility to support digital certificates if you are using a SAS Workload Orchestrator grid or a Hadoop cluster and the grid or cluster has TLS/SSL enabled.

New Grid Provider

In SAS 9.4M6, a new grid provider, provided by SAS, has been added. SAS Grid Manager consists of SAS Workload Orchestrator and SAS Job Flow Scheduler. SAS Grid Manager uses queues, host parameters, and user-specified limits to distribute jobs among hosts on the grid. High availability is supported through failover capability for the master host and defined services. The grid is configured and managed through the SAS Workload Orchestrator web interface.

Support for Grid Options Sets

A grid options set is a collection of grid options, SAS options, and required resources that are associated with a particular SAS client application. The purpose of a grid options set is to enable a SAS grid administrator to define a collection of options in SAS metadata. These options map to one or more SAS client applications and are automatically applied to workloads submitted to the grid based on the identity of the user accessing the client application. Some of the client applications that support grid options sets include the following:

- SAS Data Integration Studio
- SAS Grid Manager Client Utility
- SAS sessions that use the grdsvc_enable statement to submit jobs to the grid
- Grid-launched servers

Support for Grid-Launched Servers

With grid-launched servers, when an application requests a SAS workspace server from the object spawner, the spawner passes the request on to SAS Grid Manager. SAS Grid Manager checks the machines on the grid, using the grid's policy management to determine which host is the least busy and starts the server.

Because load-balanced servers are started by the grid provider, jobs on these servers appear as grid jobs.

In SAS 9.4M1, stored process servers and pooled workspace servers can also be grid-launched.

New Options for the SAS Grid Manager Client Utility

The following new options were added to the SAS Grid Manager Client Utility (SASGSUB):

-GRIDOPTSET
: specifies the name of a grid options set.

-GRIDWAITTIMEOUT
: specifies that the SAS Grid Manager Client Utility waits until either the job has completed running or a specified time-out value is reached.

-GRIDWAITRESULTS
: specifies that the SAS Grid Manager Client Utility waits until the job has completed running (either successfully or with an error) and then returns the results of the processing.

PROMPT
: when used as a value for the METAUSER or GRIDUSER parameters, specifies that the SAS Grid Manager Client Utility prompts the user for a user ID.

In SAS 9.4M1, support was added to the SAS Grid Manager Client Utility to support interactive mode. These options enable you to start an interactive SAS session on the grid, using either SAS Display Manager mode or UNIX line mode, or to run an arbitrary command. It also enables you to monitor the output from SAS or a command running in batch mode.

-GRIDRUNSASLM
: specifies that the SASGSUB command runs a SAS session in line mode and that standard output and standard error are displayed by SASGSUB. SASGSUB also redirects standard input to the SAS session running on the grid. The grid must be on UNIX.

-GRIDRUNSASDMS
: specifies that the SASGSUB command runs a SAS session in SAS Display Manager mode. Standard output and standard error are displayed by SASGSUB, and the X Windows session is directed to the host and port specified on the -GRIDRUNSASDMS argument. The grid must be on UNIX.

-GRIDRUNCMDINT
: specifies that an arbitrary command is run on the grid in interactive mode. Standard output and standard error are displayed by SASGSUB. Standard input is directed to the program running on the grid.

-GRIDWATCHOUTPUT
: when SASGSUB is used for batch processing, specifies that the log output and the list output of the batch job are displayed on the user's machine.

In SAS 9.4M3, options were added to the SAS Grid Manager Client Utility to control whether the job directory, job log, and output listing are retained or deleted after processing.

-GRIDWAITLOGLIST
: specifies that, after a job has completed running, and after the job log and output listing are obtained, the job directory is deleted.

-GRIDWAITNORESULTS
: specifies that, after a job has completed running and the results of the processing are returned, the job log, output listing, and job directory are deleted.

-GRIDWAITNORESULTSNOLOG
: specifies that, after a job has completed running and the results of the processing are returned, the job log, output listing, and job directory are deleted, and the log and list files are not copied to the current directory.

In SAS 9.4M5, an option was added to the SAS Grid Manager Client Utility to submit a SAS program that exists on a shared file system and that therefore is shared across all grid nodes, rather than on an internal file system and accessible only by one host. Because the program is already on a shared file system, it does not have to be copied to a grid share when it runs.

-GRIDRUNPGM
: specifies the path and file name of a program that you want to run on the grid. The program is stored on a grid node machine, and is not accessible to users on a client machine.

In SAS 9.4M7, options were added to the SAS Grid Manager Client Utility to support digital certificates if you are using either:

- a SAS Workload Orchestrator grid with TLS/SSL enabled
- a Hadoop cluster with TLS/SSL enabled

-SSLCALISTLOC
: specifies the location of a single file that contains the public certificates for all of the trusted certificate authorities (CA) in the trust chain.

-SSLCERTISS
: specifies the name of the issuer of the digital certificate that should be used by TLS.

-SSLCERTLOC
: specifies the location of a file that contains a digital certificate for the machine's public key.

-SSLCERTSERIAL
specifies the serial number of the digital certificate that should be used by TLS.

-SSLCERTSUBJ
specifies the subject name of the digital certificate that should be used by TLS.

-SSLCLIENTAUTH
specifies that the server should perform client authentication.

-SSLCRLCHECK
specifies that certificate revocation lists (CRLs) are checked when digital certificates are validated.

-SSLCRLLOC
specifies the location of a file that contains a CRL.

-SSLPKCS12LOC
specifies the location of the PKCS #12 DER encoding package file that contains the certificate and the private key.

-SSLPKCS12PASS
specifies the password that TLS requires in order to decrypt the PKCS #12 DER encoding package file.

-SSLPVTKEYLOC
specifies the location of the file that contains the private key that corresponds to the digital certificate that was specified by using the SSLCERTLOC= option.

-SSLPVTKEYPASS
specifies the password that TLS requires in order to decrypt the private key.

New Job Options

The following job options that are supported by Platform Suite for SAS were added:

app
specifies the LSF application name to use.

maxpendingtime
specifies how long a job waits in the pending state before being terminated.

jobname
specifies a job name to be used for all jobs.

SAS Grid Manager for Hadoop

In SAS 9.4M3, SAS Grid Manager for Hadoop was added. 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. All of the existing SAS Grid syntax, submission modes, and integration with other SAS products and solutions is supported by SAS Grid Manager for Hadoop.

SAS Grid Manager for Platform Agent Plug-in for SAS Environment Manager

In SAS 9.4M3, a SAS Grid Manager for Platform agent plug-in was added to SAS Environment Manager. The plug-in monitors your SAS Grid Manager for Platform grid resources over time and generates events and alerts based on the monitored information.

The SAS Grid Manager for Platform agent plug-in for SAS Environment Manager provides metric data for the grid cluster and for individual grid hosts, and uses that data to perform these functions:

- display the current state of grid resources
- graph the data over time, providing a historical view and enabling you to see how the data changes
- create alerts that notify you whenever a selected measurement reaches a selected state

In SAS 9.4M5, the agent plug-in also provides metric data for grid queues.

In SAS 9.4M6, the name of the plug-in has been changed from "SAS Grid Manager agent plug-in" to "SAS Grid Manager for Platform agent plug-in".

SAS Grid Manager Agent Plug-in for SAS Environment Manager

In SAS 9.4M6, a SAS Grid Manager agent plug-in for SAS Environment Manager has been added. It provides the same functions as the SAS Grid Manager for Platform agent plug-in, but for a SAS Grid Manager environment. SAS Grid Manager consists of SAS Workload Orchestrator and SAS Job Flow Scheduler.

SAS Grid Manager Module for SAS Environment Manager

In SAS 9.4M3, a SAS Grid Manager module was added to SAS Environment Manager.

The SAS Grid Manager module enables you to configure and perform actions on grid resources and high availability applications, and provides a view of your grid at the current moment in time. The functions provided by the module are divided into these areas:

Monitoring

- perform actions on grid resources (such as opening and closing hosts and activating and deactivating queues)
- view information for the cluster, grid hosts, queues, jobs, high-availability applications, and audit logs
- manage high availability applications

In 9.4M5, these additional functions were added to manage jobs:

- terminate and requeue jobs
- switch an uncompleted job to a different queue
- change the position of jobs in the queue

LSF Configuration Manager
modify the configuration information for the LSF cluster, including definitions for hosts, queues, users, administrators, resource limits, and cluster parameters

HA Configuration Manager
modify the configuration information for high availability applications

In SAS 9.4M5, functions were added to specify dependencies and affinity requirements. A dependent application is one that either is started or continues to run according to the state of the HA application or must be started before the HA application starts. An affinity requirement specifies where a selected application runs in relation to other applications in the HA configuration.

Utilities, added in 9.4M5 (available for LSF administrator only)
update the LSF license

Note: The SAS Grid Manager module for SAS Environment Manager replaces the SAS Grid Manager server plug-in that was added to SAS Environment Manager in SAS 9.4M2.

New Functions

In SAS 9.4M3, these functions were added:

GRDSVC_HOSTLIST
: This function returns a list of hosts that are available to run grid jobs.

GRDSVC_OPTSETS
: Returns a list of valid options sets for a specified combination of SAS Application Server, SAS client application, and user.

Shortened SAS Grid Manager Client Utility Option Names

In SAS 9.4M5, alternate shortened forms of most SAS Grid Manager Client Utility options were added. Both the long and short versions of the option name are valid. In addition, several options have multiple valid forms. Here are the shortened option names:

Original Form	Shortened Forms
GRIDAPPSERVER	APPSERVER, SERVER
GRIDFILESIN	FILESIN
GRIDFORCECLEAN	FORCECLEAN
GRIDGETRESULTS	GETRESULTS
GRIDGETSTATUS	GETSTATUS
GRIDJOBNAME	JOBNAME
GRIDJOBOPTS	JOBOPTS
GRIDKILLJOB	KILLJOB
GRIDLICENSEFILE	LICENSEFILE, LICENSE
GRIDLRESTARTOK	LRESTARTOK, LRESTART

Original Form	Shortened Forms
GRIDOPTSET	OPTSET
GRIDPASSWORD	GRIDPASS, GRIDPWD
GRIDPLUGINPATH	PLUGINPATH
GRIDRESTARTOK	RESTARTOK, RESTART
GRIDRESULTSDIR	RESULTSDIR
GRIDRUNCMD	RUNCMD
GRIDRUNCMDINT	RUNCMDINT
GRIDRUNPGM	RUNPGM
GRIDRUNSASDMS	RUNSASDMS
GRIDRUNSASLM	RUNSASLM
GRIDSASOPTS	SASOPTS
GRIDSETPERMS	SETPERMS
GRIDSTAGECMD	STAGECMD
GRIDSTAGEHOST	STAGEHOST
GRIDSUBMITPGM	SUBMITPGM
GRIDWAIT	WAIT
GRIDWAITLOGLIST	WAITLOGLIST
GRIDWAITNORESULTS	WAITNORESULTS
GRIDWAITRESULTS	WAITRESULTS
GRIDWAITRESULTSNOLOG	WAITRESULTSNOLOG
GRIDWAITTIMEOUT	WAITTIMEOUT, TIMEOUT
GRIDWATCHOUTPUT	WATCHOUTPUT
GRIDWORKLOAD	WORKLOAD
HELP	H, ?

Original Form	Shortened Forms
VERBOSE	TRACE

PART 1

Introduction to SAS Grid Computing

1

What Is SAS Grid Computing?

SAS Grid Computing Basics

A SAS grid computing environment is one in which SAS computing tasks are distributed among multiple computers on a network, all under the control of SAS Grid Manager. In this environment, workloads are distributed across a grid cluster of computers. This workload distribution enables the following functionality:

Workload balancing
: enabling multiple users in a SAS environment to distribute workloads to a shared pool of resources.

Accelerated processing
: allowing users to distribute subtasks of individual SAS jobs to a shared pool of resources. The grid enables the subtasks to run in parallel on different parts of the grid, which completes the job much faster.

Scheduling jobs
: allowing users to schedule jobs, which are automatically routed to the shared resource pool at an appropriate time.

SAS Grid Manager provides load balancing, policy enforcement, efficient resource allocation, prioritization, and a highly available analytic environment for SAS products and solutions running in a shared grid environment. It also separates the SAS applications from the infrastructure used to execute the applications. This enables you to transparently add or remove hardware resources as needed and also provides tolerance of hardware failures within the grid infrastructure. SAS Grid Manager integrates the resource management and scheduling capabilities of the grid provider with the SAS 4GL syntax and subsequently with several SAS products and solutions.

SAS Grid Topology

SAS Grid Configuration

A SAS Grid configuration consists of these main components:

Grid control server
: this machine controls distribution of jobs to the grid. Any machine on the grid can be designated as the grid control server. You can also choose whether to configure the grid control server as a grid resource capable of receiving work. This machine must contain Base SAS and SAS/CONNECT. The grid control server might also configure a SAS workspace server so that SAS applications (SAS Data Integration Studio, SAS Enterprise Miner, SAS Enterprise Guide, and SAS Add-In for Microsoft Office) can run programs that take advantage of the grid.

Grid node
: these machines are grid computing resources that are capable of receiving the work that is being distributed to the grid. The number of nodes in a grid depends on the size, complexity, and volume of the jobs that are run by the grid. You can add or remove nodes as specified by your business needs. Each grid node must contain Base SAS, SAS/CONNECT, and any applications and solutions needed to run grid-enabled jobs.

Shared file system
: this component is used to store data and shared directories for jobs that run on the grid. In order to simplify installation and ease maintenance, you can also install the SAS binaries on the shared file system if your grid runs on UNIX machines.

Metadata server
: this machine contains the metadata repository that stores the metadata definitions needed by all SAS Grid Manager products as well as other SAS

applications and solutions that are running on the grid. Although it is recommended that the SAS Metadata Server be on a dedicated machine, it can be run on the grid control server.

SAS Management Console
this application is used to manage the definitions in the metadata repository and to submit jobs to the grid through the Schedule Manager plug-in

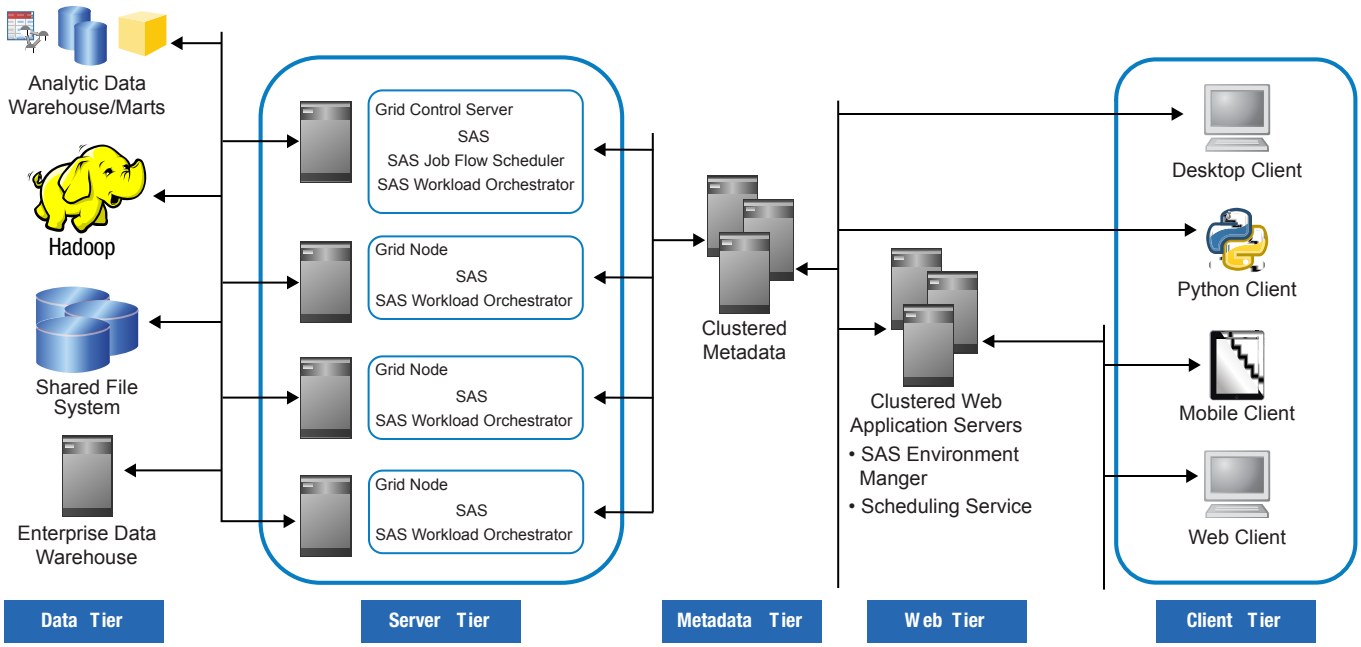
Grid administration application
an application or a component of an application used to monitor and manage grid resources. The application that you use depends on the application that is used to provide grid services. A SAS Grid Manager grid uses the SAS Workload Orchestrator web interface and an agent plug-in for SAS Environment Manager for administration and monitoring. SAS Grid Manager for Platform, which uses Platform Suite for SAS for grid services, uses the SAS Grid Manager Module for SAS Environment Manager and the SAS Environment Manager agent plug-in for administration and monitoring.

Grid clients
clients submit jobs to the grid for processing, but they are not part of the grid resources that available to execute work. Examples of grid clients are:

- a SAS Data Integration Studio client, a SAS Enterprise Miner client, or a SAS Enterprise Guide client that uses a workspace server that can be spawned by the grid
- a SAS Management Console client that uses the Schedule Manager plug-in or another application to schedule SAS workflows
- a SAS Foundation install (SAS Display Manager) that is used to run a program that submits work to the grid. The submitted work can either be entire programs or programs that are divided into parallel chunks. Base SAS and SAS/CONNECT must be installed on this client.
- a SAS Grid Manager Client Utility (SASGSUB). SAS is not required to be installed on this client.

Grid Architecture with SAS Grid Manager

The following diagram is a conceptual view of the SAS Grid Manager architecture when using SAS Workload Orchestrator and SAS Job Flow Scheduler.



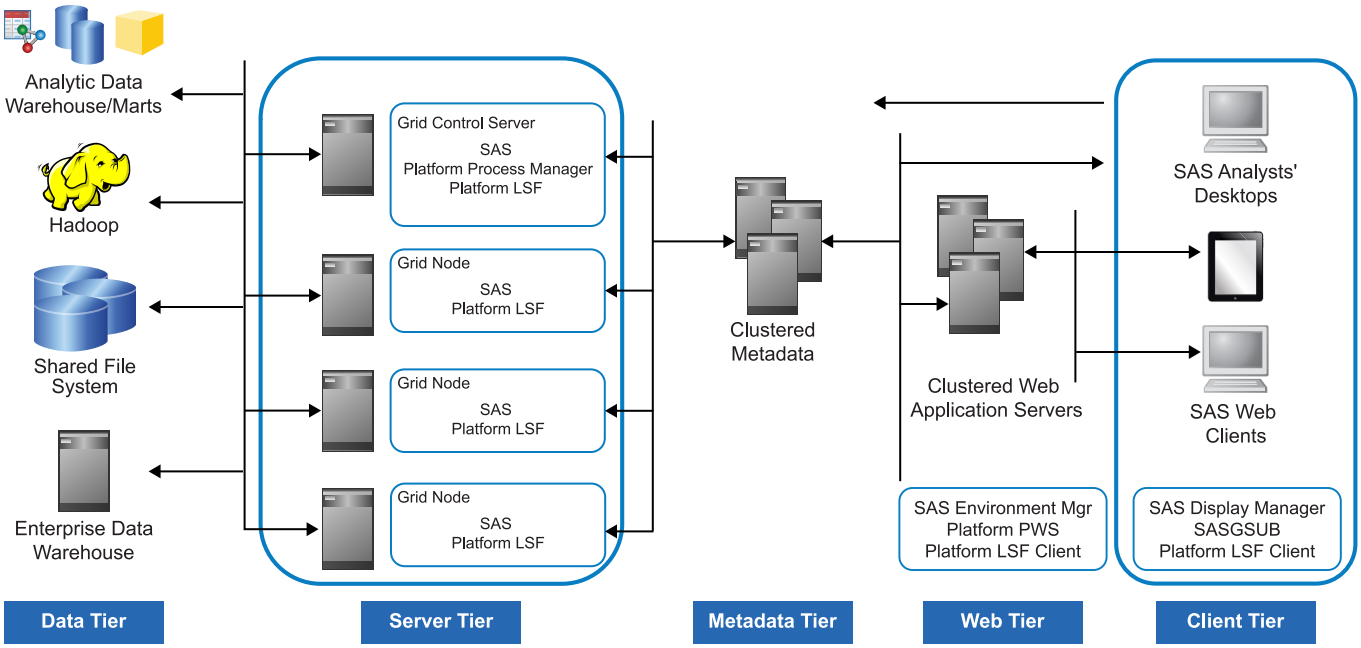
SAS Grid Manager uses the following components:

- The purpose of the hosts that run the SAS Workload Orchestrator daemon is to run jobs and services. The master host orchestrates the jobs and services. Hosts can be designated as master host candidates. One master candidate serves as the master, and other master candidates are designated to assume the role of master host if the original master host fails. The master host controls the scheduling of jobs that are in queues to the appropriate hosts. The master host also keeps track of the states of all other hosts. The master host also determines which hosts are used to run services.
- SAS Job Flow Scheduler schedules jobs to run on the grid.
- SAS Workload Orchestrator web interface provides an integrated facility for configuring, monitoring, and managing grid resources.

Grid Architecture with SAS Grid Manager for Platform

The following diagram is a conceptual view of the SAS grid architecture when using SAS Grid Manager for Platform, which uses Platform Suite for SAS. Platform Suite for SAS is a set of components that provide efficient resource allocation, policy management, and load balancing of SAS workload requests.

Figure 1.1 *Grid Architecture Diagram — SAS Grid Manager for Platform*



Platform Suite for SAS includes these components:

Load Sharing Facility (LSF)
: This facility dispatches all jobs submitted to it, either by Process Manager or directly by SAS, and returns the status of each job. LSF also manages any resource requirements and performs load balancing across machines in a grid environment. A server version of Platform LSF must be on the Grid Control Server and each Grid Node. A client version of Platform LSF must be installed with Platform PWS and with the SAS Display Manager and SASGSUB grid clients. Platform LSF is not required for any other grid clients.

Process Manager (PM)
: This is the interface used by the SAS scheduling framework to control the submission of scheduled jobs to LSF and to manage any dependencies between the jobs. Process Manager includes two optional components, Calendar Editor and Flow Manager. Calendar Editor is a scheduling client for a Process Manager server. It enables you to create new calendar entries for time dependencies. Flow Manager provides a visual representation of flows that are created and scheduled through the Schedule Manager plug-in for SAS Management Console as well as reports scheduled through SAS Web Report Studio. Flow Manager enables you to view and update the status of jobs in a flow and rerun jobs. Platform PM is typically installed on the Grid Control Server.

Platform Web Services (PWS)
: This is a REST-based web service that provides information to the SAS Grid Manager for Platform module and agent plug-in available in SAS Environment Manager. It requires that an LSF client be installed on the same machine.

Enterprise Grid Orchestrator (EGO)
: Provides support for critical service failover in a highly available (HA) environment. EGO provides functions for monitoring critical services, restarting services if they stop, and starting services on a failover host when needed. EGO is a component of LSF and not explicitly shown on the diagram above. However, LSF must be installed on any machine that is running a service for which EGO is to provide high availability.

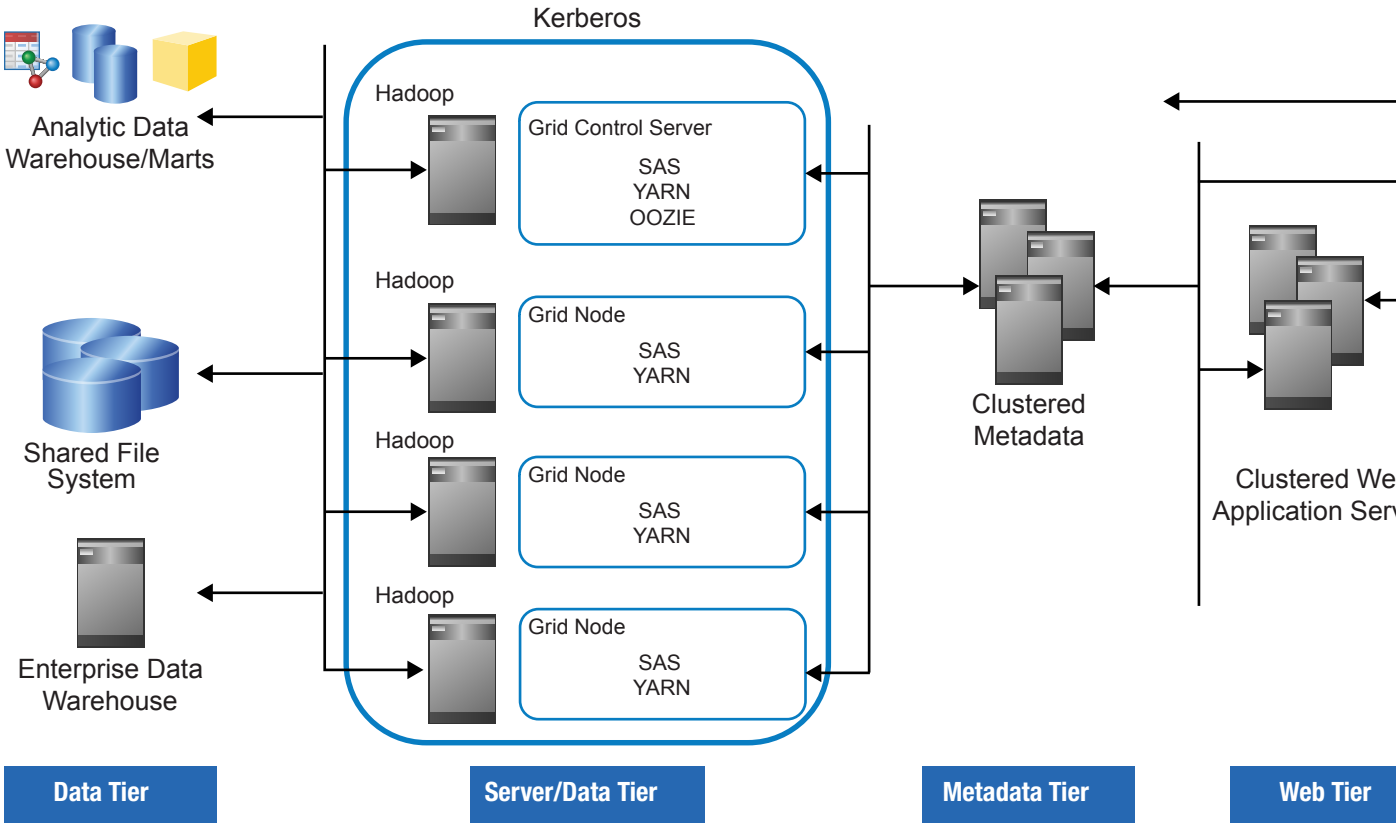
Platform RTM for SAS
: A web-based tool in releases prior to SAS 9.4M3 that enabled you to graphically view the status of devices and services in a SAS grid environment as well as manage the policies and configuration.

> **IMPORTANT** Platform RTM 2.0.8 is not part of Platform Suite for SAS. The version of RTM that was available as a download for use with SAS Grid Manager is in Limited Support. See SAS Technical Support Services and Policies for more information about Limited Support. Contact SAS Technical Support for information about RTM.

Grid Architecture with SAS Grid Manager for Hadoop

The following diagram is a conceptual view of the SAS Grid Architecture when using SAS Grid Manager for Hadoop.

Figure 1.2 *Grid Architecture Diagram — SAS Grid Manager for Hadoop*



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. SAS Grid Manager for Hadoop uses the following components and features:

- YARN provides the resource management for the SAS Grid jobs. YARN must be installed on the grid control server and on each of the grid nodes that are expected to run SAS Grid jobs. You must configure the grid control server on the same machine on which YARN's Resource Manager is running.
- Oozie schedules SAS workflows across the cluster. The Oozie server must be installed on one of the nodes in the cluster and is shown on the Grid Control Server in the above diagram.
- Kerberos must be configured for the Hadoop cluster. This is required in order for YARN to be able to run the SAS Grid jobs with the identity of the user that submitted them.
- There are no Hadoop components required on the SAS web tier. All monitoring and management of SAS Grid jobs on the Hadoop cluster is performed through the usual Hadoop monitoring and management tools.
- the SAS clients require the Hadoop JAR and configuration files that are installed by the SAS Deployment Wizard and SAS Deployment Manager.

The need for a shared file system is greatly reduced since most data will be stored in HDFS. However, HDFS is not a POSIX compliant file system, so you will still need to use a shared file system rather than HDFS for the following:

- the shared directory required by SASGSUB
- installation and configuration of SAS. If you do not want to use a shared file system for installing and configuring SAS, you can use parcels to leverage Cloudera Manager or packages to leverage Ambari.
- shared project and data directories that are required by other SAS products running on the grid, such as SAS Enterprise Miner and SAS Forecast Server

What Types of Processing Does a Grid Support?

Multi-User Workload Balancing

Most organizations have many SAS users performing a variety of query, reporting, and modeling tasks and competing for the same resources. SAS Grid Computing can provide order to this environment by providing capabilities such as the following:

- specifying which jobs get priority
- deciding the share of computing resources used by each job
- controlling the number of jobs that are executing at any one time

In practice, SAS Grid Computing acts as a gatekeeper for the jobs that are submitted to the grid. As jobs are submitted, SAS Grid Computing dispatches the job to grid nodes, preventing any one machine from being overloaded. If more jobs are submitted than can be run at once, SAS Grid Computing submits as many jobs as can be run. The rest of the jobs are held in a queue until resources are free, and

then the jobs are dispatched to be run. SAS Grid Computing can also use job priority to determine whether a job is run immediately or held in a queue.

The application user notices little or no difference in their processes when working with a grid. For example, users can define a key sequence to submit a job to a grid rather than running it on their local workstation. Batch jobs can be run using the SAS Grid Manager Client Utility.

Parallel Workload Balancing

Some SAS programs consist of subtasks that are independent units of work and can be distributed across a grid and executed in parallel. You can use SAS syntax to identify the parallel units of work in these programs, and then use SAS Grid Computing to distribute the programs across the grid. Using parallel workload balancing can substantially accelerate the entire application.

Applications such as SAS Data Integration Studio, SAS Risk Dimensions, and SAS Enterprise Miner are often used for iterative processing. In this type of processing, the same analysis is applied to different subsets of data or different analysis is applied to a single subset of data. Using SAS Grid Computing can improve the efficiency of these processes, because the iterations can be assigned to different grid nodes. Because the jobs run in parallel, the analysis completes more quickly and with less strain on computing resources.

Distributed Enterprise Scheduling

The Schedule Manager plug-in for SAS Management Console provides the ability to schedule user-written SAS programs as well as jobs from numerous SAS applications. You can schedule the jobs and programs to run when specified time or file events occur. The jobs are then run on the grid using the resource and prioritization policies established by SAS Grid Manager.

You can also use SAS Grid Manager Client Utility to interface SAS Grid Manager with an existing enterprise scheduler so that scheduled jobs are processed by the grid without any change in how the scheduler is used.

SAS Applications That Support Grid Processing

The following table lists the SAS applications that currently support grid processing and the type of processing that each supports.

Note: Starting with SAS 9.4M8, JMP Genomics is not available from SAS. If you have an existing installation of JMP Genomics in your environment and plan to upgrade to JMP Pro 17, please contact JMP Technical Support for assistance.

Table 1.1 *Grid Support in SAS Applications*

SAS Application	Multi-User Workload Balancing	Parallel Workload Balancing	Distributed Enterprise Scheduling
Any SAS program	yes	yes, with modifications	yes
SAS Grid Manager Client Utility (SASGSUB)	yes	yes	yes
SAS Enterprise Guide	yes	yes	
SAS Add-In for Microsoft Office	yes	yes	
SAS Data Integration Studio	yes	yes	yes
SAS Enterprise Miner	yes	yes	
SAS Risk Dimensions	yes	yes, with modifications	
SAS Web Report Studio			yes
SAS Marketing Automation			yes
SAS Marketing Optimization			yes
SAS JMP Genomics		yes	
SAS Demand Forecasting for Retail		yes	
SAS products or solutions that use workspace server load balancing	yes		
SAS stored processes	yes, with limitations	yes, with limitations	

What Business Problems Can a Grid Solve?

Many Users on Single Resource

An organization might have multiple users submitting jobs to run on one server. When the environment is first configured, the server might have been sufficient to handle the number of users and jobs. However, as the number of users submitting jobs grows, the load on the server grows. The increased load might lead to slower processing times and system crashes. In a SAS grid environment, jobs are automatically routed to any one of the servers on the grid. This spreads the computing load over multiple servers, and diminishes the chances of a server becoming overloaded. If the number of jobs exceeds the resources available, the jobs are queued until resources become available. If the number of users continues to increase, you can increase capacity by adding servers to the grid.

High Availability

Your organization might have services and long-running SAS programs that are critical to your operations. The services must be available at all times, even if the servers that are running them become unavailable. The SAS programs must complete in a timely manner, even if something happens to cause them to fail. For a SAS program that takes a long time to run, this means that the program cannot be required to restart from the beginning if it ends prematurely.

You can configure the critical services within your SAS grid environment to be highly available. SAS Grid Manager and SAS Grid Manager for Platform can monitor the critical services, detect if they fail or if the machine on which they are running fails, and automatically start the services on a failover host. This ensures that critical services remain available to clients without any manual intervention.

By using options on the SAS Grid Manager Client Utility, you can specify that SAS programs submitted to the grid are automatically restarted from the point where they stopped if they end before completion. The job restarts from the last completed procedure, DATA step, or labeled section. Jobs that take a long time to run do not have to start over at the beginning. You can also use the restart capability with queue options that automatically requeue jobs that end prematurely to provide a complete high-availability environment for SAS programs.

Increased Data Growth

Your organization might have a process running to analyze a certain volume of data. Although the server that is processing the job is sufficient to handle the current volume of data, the situation might change if the volume of data increases. As the amount of data increases, the load on the server increases, which can lead to longer processing times or other problems. Changing to a larger-capacity server can involve considerable expense and service interruption.

A SAS grid environment can grow to meet increases in the amount of data processed. If the volume of data exceeds the capacity of a server on the grid, the processing load can be shared by other grid servers. If the volume continues to increase, you can add servers to the grid without having to make configuration changes to your processes. Adding servers to the grid is also more cost-effective than replacing a single large server, because you can add smaller servers to handle incremental increases in data volume.

Running Larger and More Complex Analysis

Your organization might have a process running to perform a certain level of analysis on data. If you want to increase the complexity of the analysis being performed, the increased workload puts a greater strain on the processing server. Changing the computing power of the server involves considerable expense and interrupts network availability.

Using a SAS grid environment enables you to add computing power by adding additional computers on the grid. The analysis job can be divided up among the grid nodes, which enables you to perform more complex analysis without increasing the load on any single machine.

Need for a Flexible IT Infrastructure

Your organization's ability to perform the data analysis that you need depends on a flexible computing infrastructure. You must be able to add needed resources quickly and in a cost-effective manner as the load increases. You must also be able to handle maintenance issues (such as adding or replacing resources) without disrupting your work. A SAS grid environment enables you to maintain a flexible infrastructure without disrupting your operations.

As your data-processing needs grow, you can incrementally add computing resources to your grid by adding smaller, less-expensive servers as new server nodes. This ability prevents you from having to make large additions to your environment by adding large and expensive servers.

When you need to perform maintenance on machines on the grid, the grid can still operate without disruption. When you take the servers offline for maintenance or upgrades, SAS Grid Computing routes to work to the machines that are still online.

Users who send work to the grid for processing do not have to change their way of working. Work that is sent to the grid is processed just as before.

Likewise, the SAS grid environment adapts if a computer fails on the grid. Because SAS Grid Manager automatically avoids sending work to the failed machine, the rest of the grid is still available for processing and users do not see any disruption.

PART 2

Grid Computing for SAS Using SAS Grid Manager

2

Planning and Configuring a SAS Grid Manager Environment

SAS Workload Orchestrator Installation and Configuration Overview

The process of installing and configuring a SAS Workload Orchestrator grid is integrated with the process of installing and configuring SAS products and metadata definitions on the grid. You can either install all SAS products on all machines in the grid or install different sets of SAS applications on sets of machines in the grid. However, Base SAS, SAS/CONNECT, and SAS Workload Orchestrator must be installed on all grid machines. Using a grid plan file with the SAS Deployment Wizard guides you through the process of installing and configuring the SAS applications and metadata definitions on each machine in the grid. You must specify the same SASHome and SASConfig directory structures on all machines in the grid, and you must use the same installer ID on each grid node.

For information about performing a planned installation, see SAS Intelligence Platform: Installation and Configuration Guide.

Configuring the File Server

The central file server is a critical component of a grid environment. It is essential for each application on a grid node to be able to efficiently access data. Slowdowns caused by the performance of the file storage system could reduce the effectiveness and benefits of using a grid. The amount of storage required and the type of I/O transactions help determine the type of file storage system that best meets your needs. A shared file system is required if you want to use the restart functions of the SAS Grid Manager Client Utility.

Assuming that the SAS jobs that are running on the grid perform an equal number of reads and writes, it is recommended that the file system be able to sustain 75–100

MB per second per core. This level can be adjusted up or down, depending on the level of I/O activity of your SAS jobs. For information about choosing and configuring a file system, see Shared File Systems: Determining the Best Choice for your Distributed SAS® Foundation Applications and Troubleshooting system performance problems: Shared/clustered file systems.

Installing SAS Grid Manager

SAS Grid Manager is installed as part of the SAS installation process. The SAS Deployment Wizard installs and configures these components on the grid control server:

Table 2.1 *SAS Deployment Wizard Grid Control Server Components*

Installed SAS Software Components	Configured SAS Software Components
■ SAS Foundation (including Base SAS and SAS/CONNECT) ■ SAS Workload Orchestrator ■ SAS Workload Orchestrator Administration Utility (optional) ■ SAS Job Flow Scheduler ■ SAS Management Console ■ SAS Environment Manager (can also be installed on a different machine)	■ Web Infrastructure Platform Scheduling Services ■ SAS Workload Orchestrator ■ SAS Job Flow Scheduler Server ■ SAS Environment Manager Server (can also be on a separate machine) ■ SAS Grid Manager Agent Plug-in for SAS Environment Manager (can also be on a separate machine) ■ SAS Application Server (SAS Logical DATA Step Batch Server, SAS Logical Grid Server, SAS Logical Workspace Server, SAS Workspace Server) ■ SAS Metadata Server (can also be on a separate machine) ■ Object Spawner

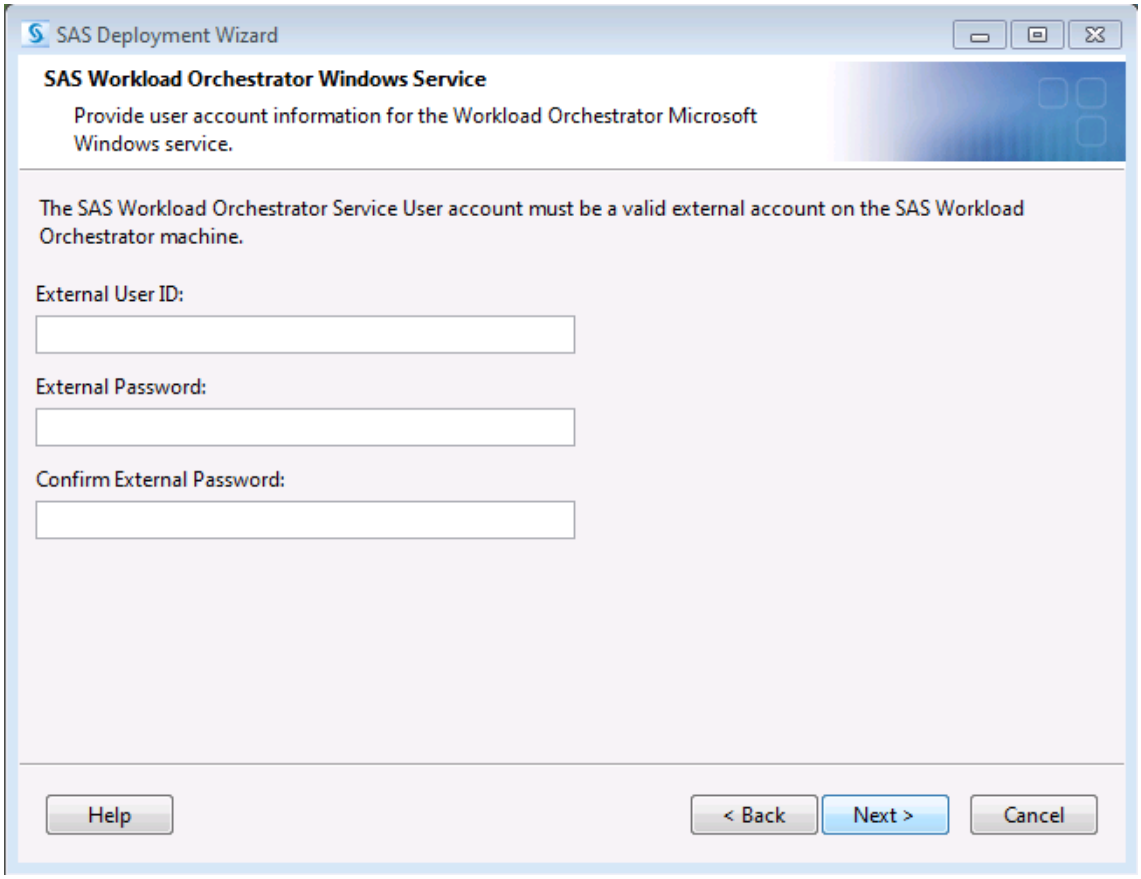
The amount of user input that is required during the installation and configuration process depends on whether you choose an Express, Typical, or Custom installation. For information about running the SAS Deployment Wizard, see *SAS Intelligence Platform: Installation and Configuration Guide*.

An Express installation requests only the shared directory (see Step 4) and SAS Workload Orchestrator grid information (see Step 5). Because default values are used in all other cases, you must verify that these values match the values that are needed for your environment.

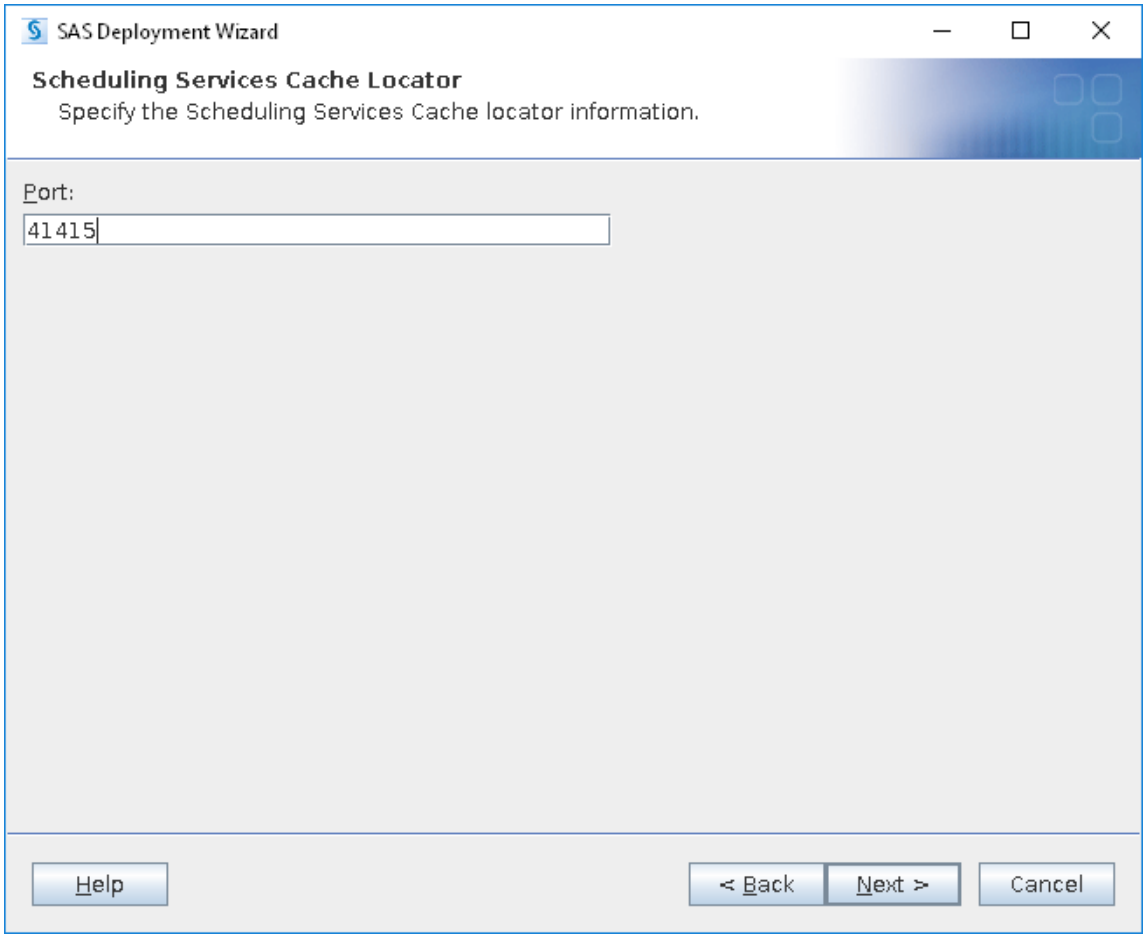
During the installation and configuration process for a Custom installation or a Typical installation, the SAS Deployment Wizard displays these pages that request grid-specific information:

1 If you are installing on Windows, the SAS Workload Orchestrator Windows Service page enables you to specify the account that the SAS Workload Orchestrator daemon runs under when it runs as a Windows service. You must specify a fully qualified domain account so that SAS Workload Orchestrator can access the shared file system in case of a master failover. This account must be a local administrator (a member of the administrators group on the machine). The user must have “Logon as service” and “Replace process level token” user rights.

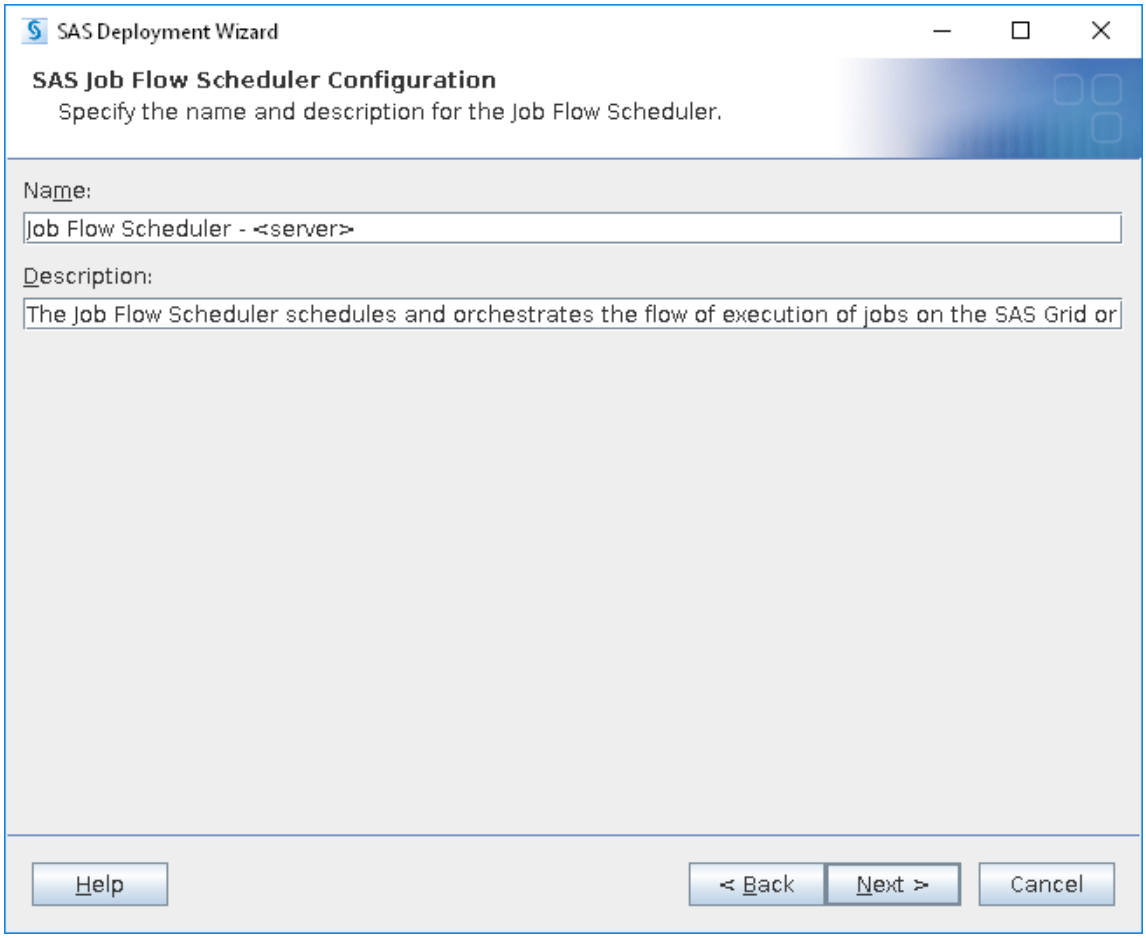
If you are installing on UNIX, this step does not apply.



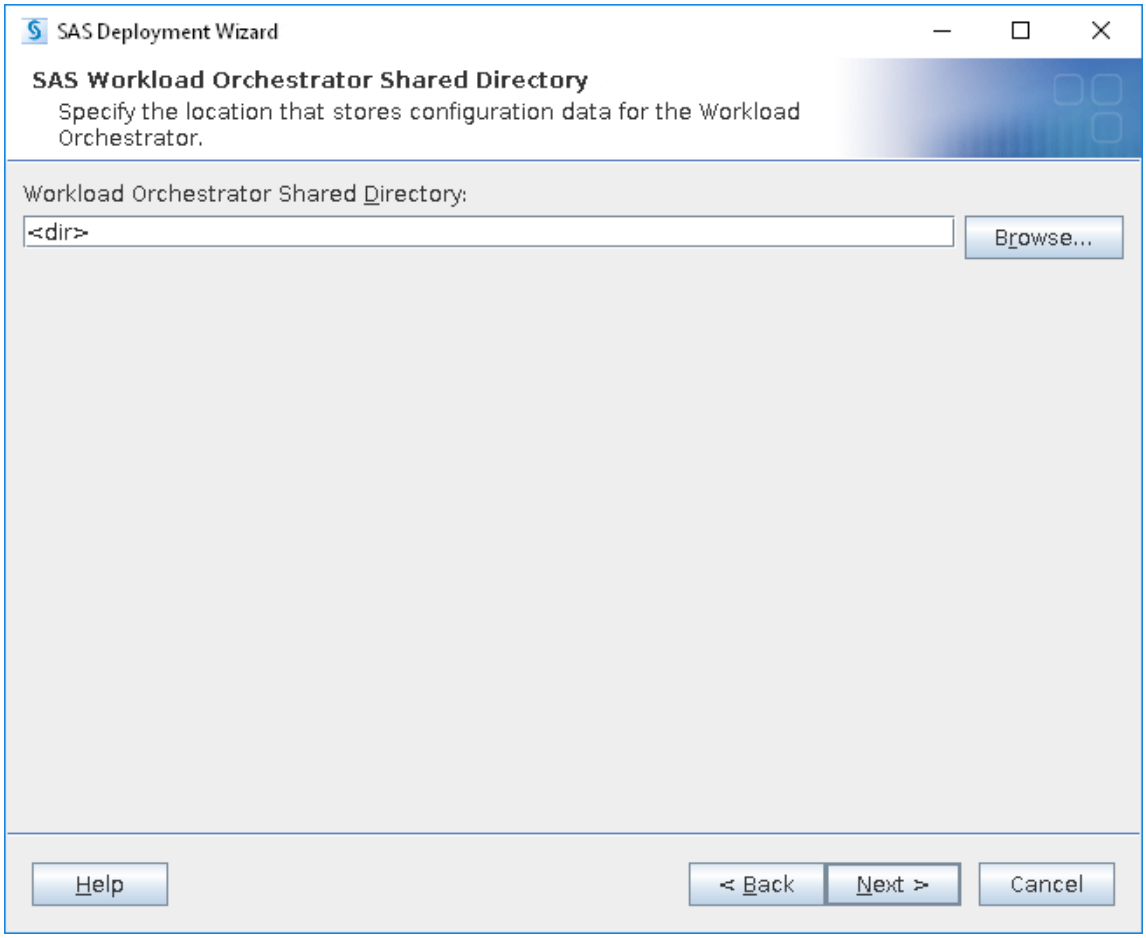
2 The Scheduling Services Cache Locator page enables you to specify the port to connect to the host that serves as a cache location during job scheduling.



3 The SAS Job Flow Scheduler Configuration page enables you to name the host that runs the Job Flow Scheduler, which schedules and orchestrates the flow of execution of jobs on the SAS Workload Orchestrator grid or directly on the operating system. The host machine is specified in the plan file.

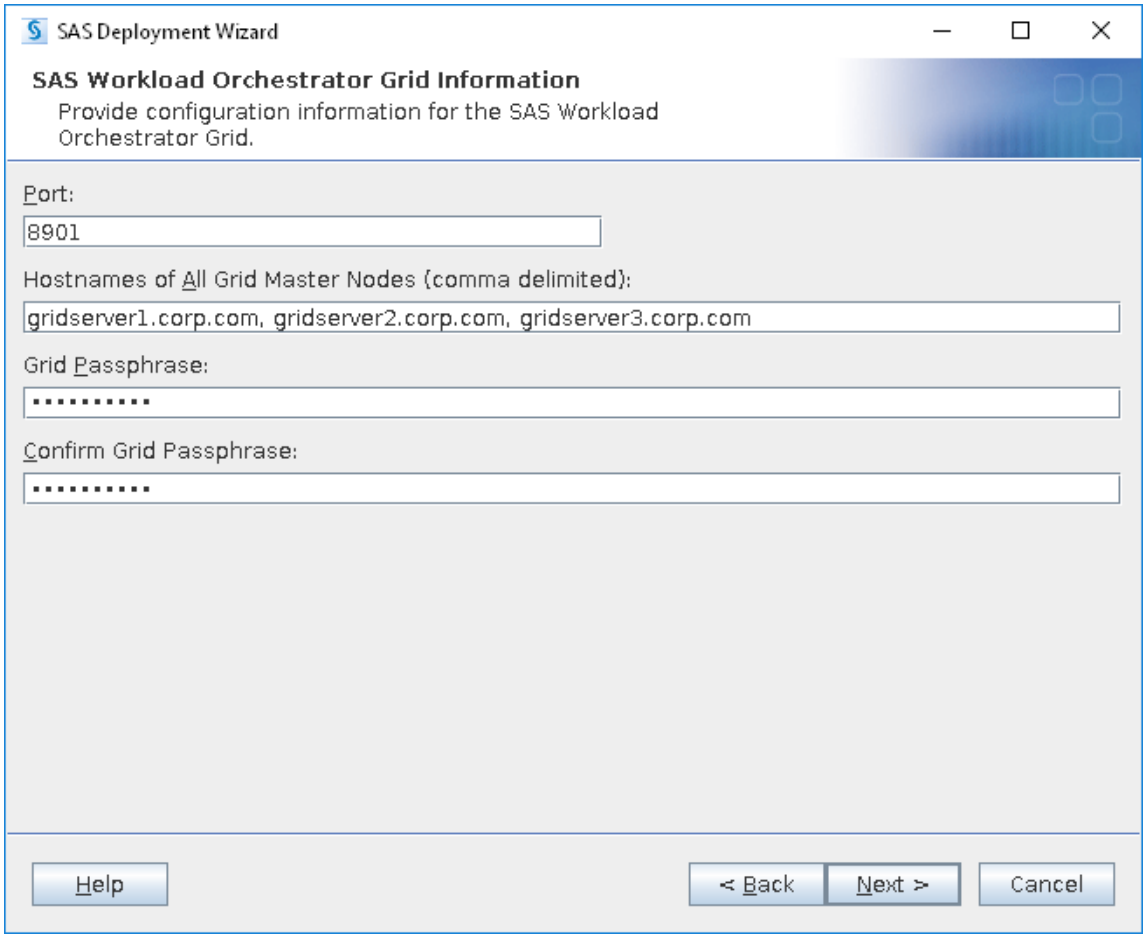


4 The SAS Workload Orchestrator Shared Directory page enables you to specify the shared directory that contains the configuration data for the grid. If the master host fails and a new master takes over, it reads the configuration data in the shared location in order to determine the configuration of the grid. Specify a fully qualified pathname to which all nodes in the grid have access.



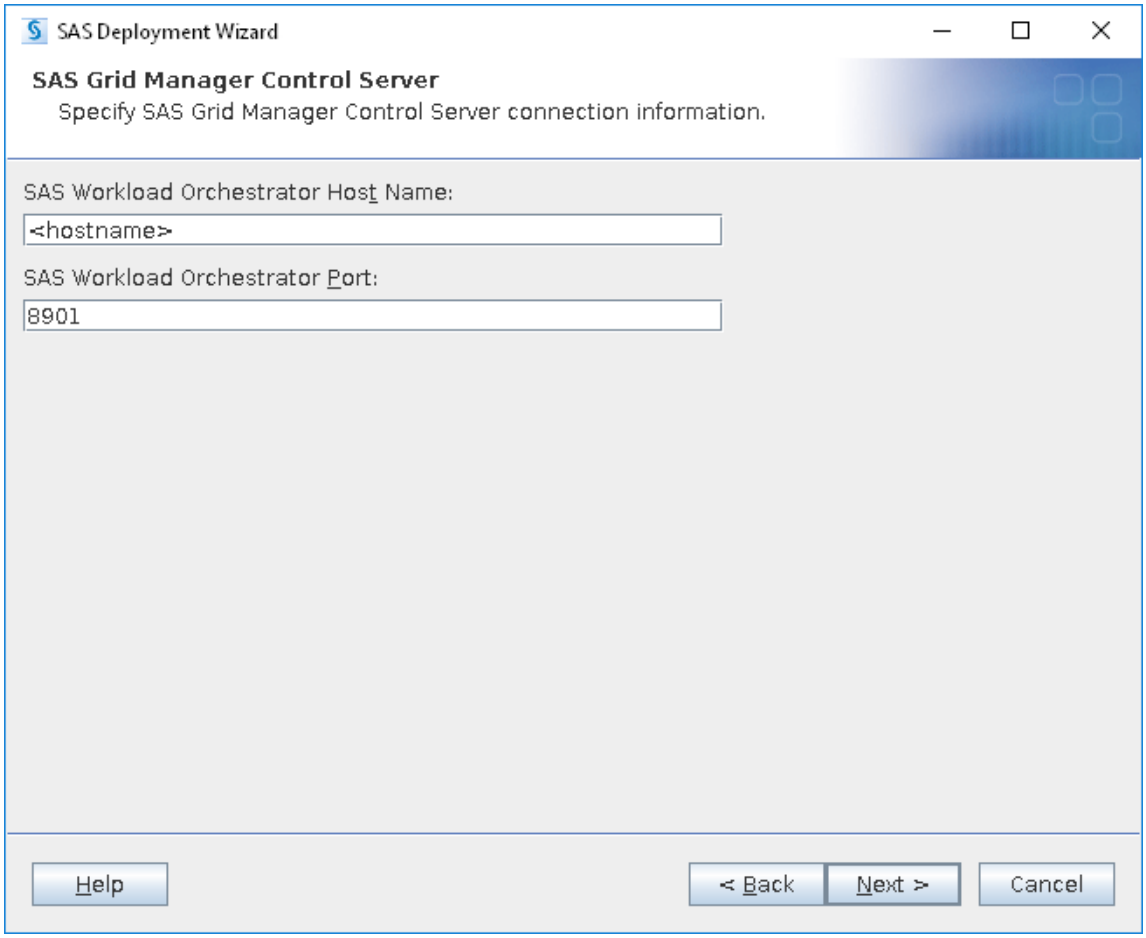
5 The SAS Workload Orchestrator Grid Information page enables you to specify the port for the master host and a list of hosts that are candidates to serve as the master host for the grid. Specify the hosts using fully qualified domain names, separated by commas. If the current master host fails, the other master candidates in the list can poll each other in the order that is specified in the **Hostnames of All Grid Master Nodes** field. The first host to respond takes over as the master.

The passphrase protects sensitive information that is on the grid.



6 The SAS Grid Manager Control Server page enables you to specify the server to be used as the SAS Logical Grid Server, which is the first specified master host. In the **SAS Workload Orchestrator Host Name** field, specify the fully qualified name of the master host for the grid. It is the same as the first host that was specified in the **Hostnames of All Grid Master Nodes** field on the previous page.

Note: If you are performing a Typical installation, specify only the SAS Workload Orchestrator host name and port.

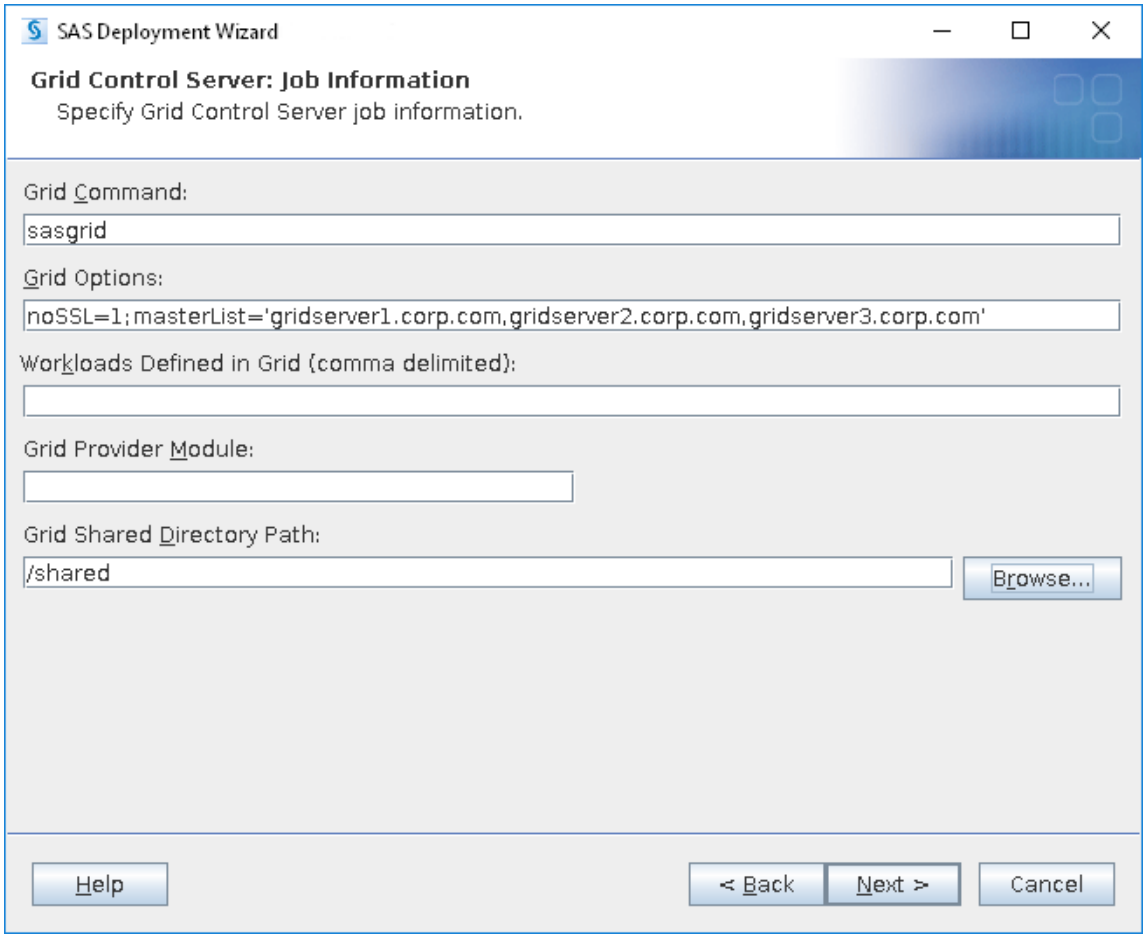


7 The SAS Grid Manager Control Server: Job Information page enables you to specify how jobs run on the grid. Specify the command that is executed on the grid node. Specify options for connecting to the grid and for running jobs on the grid. You can also specify workloads (tags) defined in the grid, the grid provider module (usually left blank), and the shared directory for the grid. If you are performing a Typical installation, you must specify only the shared directory.

By default, the **Grid Options** field specifies that SSL is not used, and it lists all the possible master hosts that you specified in step 5.

The directory in the **Grid Shared Directory Path** field is used by the SAS Grid Manager Client Utility if a shared directory is not used between the grid and the client. By default, this directory is set to the shared directory that is used by SAS Workload Orchestrator. However, if possible, you should use a different shared directory so that the SAS Workload Orchestrator and the SAS Grid Manager Client Utility information can be kept separate.

Do not change the values in the **Grid Command** and **Grid Provider Module** fields unless instructed to do so by SAS Technical Support.



8 Complete the installation process as prompted by SAS Deployment Wizard.

By default, the Linux SAS Workload Orchestrator daemon that runs as the SAS installer does not have permission to read all the I/O statistics of the processes that it manages. If the I/O statistics are important to your deployment, see "Enabling I/O Metrics in Linux" on page 92.

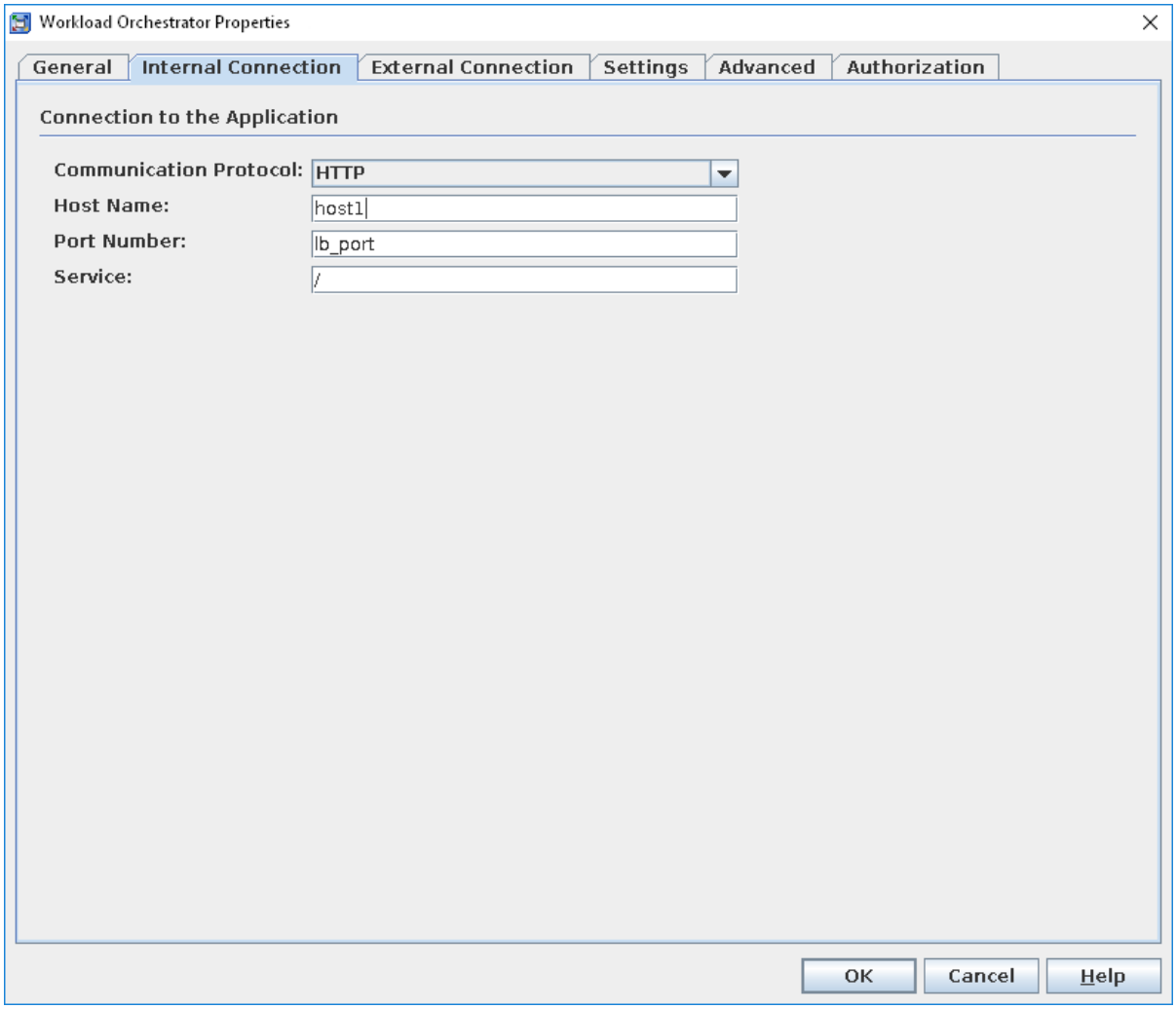
Enabling Master Host Failover for Job Flow Scheduler

SAS Workload Orchestrator provides master host failover capability using a list of master host candidates. If the current master host fails, the other master candidates in the list poll each other in the order in which they are specified. The first host to respond takes over as the master. The master host candidates are specified during installation, However, you can also add master host candidates after installation. See "Overview" on page 49 for more information.

However, the process of assigning a new master host does not update the information in metadata, which SAS Job Flow Scheduler uses to communicate with the grid. If you are creating a multi-node grid, follow these steps to use SAS Web

Server as a load balancer that directs SAS Job Flow Scheduler to the correct master host.

1. In SAS Management Console, select the **Configuration Manager** plug-in.
2. In the list of configuration entries, right-click **Workload Orchestrator** and select **Properties**.
3. Select the **Internal Connection** tab.
4. In the **Port Number** field, specify an available but unused port (referred to in this process as *lb_port*). This port must be different from the port that SAS Workload Orchestrator is using. Click **OK** to save your changes.



5. In the directory ***SASConfig*/Levn/Web/WebServer/conf**, create the file swo.conf. Here is the content of the swo.conf file:

```
#IfModule !watchdog_module
<IfModule !watchdog_module>
LoadModule watchdog_module "<SASHome>/SASWebServer/9.4/httpd-2.4/modules/mod_watchdog.so"
</IfModule>

# Load the health check module.
<IfModule !proxy_hcheck_module>
```

```
LoadModule proxy_hcheck_module "<SASHome>/SASWebServer/9.4/httpd-2.4/modules/
mod_proxy_hcheck.so"
</IfModule>

# LogLevel proxy_hcheck:TRACE8

# Specify a port for SAS Workload Orchestrator traffic.
Listen lb_port

# Configure a virtual host (using the port specified above) to route traffic
# to SAS Workload Orchstrator.
<VirtualHost *:lb_port >

  # We use a health check to ensure that only the current master is available. All other
hosts
  # will fail the health check and be disabled. Only the current master will return 401
on
  # GET /sasgrid/index.html.
  # All other hosts will return 301.
  ProxyHCExpr ok401 {%{REQUEST_STATUS} == 401 || %{REQUEST_STATUS} == 200}

  ProxyPass / balancer://SASWorkloadOrchestrator/
  ProxyPassReverse / balancer://SASWorkloadOrchestrator/

  # The health checks ensures that all traffic will be routed to the current master.
However,
  # we can optimize even further by assigning each balancer member to a balancer member
set (where
  # each set contains only a single host). This ensures that Apache will traverse the
list of
  # master candidates in the same order as SAS Workload Orchstrator.

<Proxy balancer://SASWorkloadOrchestrator>
BalancerMember http://host_candidate1:host_port lbset=0 hcinterval=5
hcmethod=GET hcuri=/login/login.html hcexpr=ok401
    BalancerMember http://host_candidate2:host_port lbset=1 hcinterval=5
hcmethod=GET hcuri=/login/login.html hcexpr=ok401
    BalancerMember http://host_candidate3:host_port lbset=2 hcinterval=5
hcmethod=GET hcuri=/login/login.html hcexpr=ok401

</Proxy>

</VirtualHost>
```

The value of *lb_port* is the same as that specified in SAS Management Console in step 4. The values of *host_candidate:host_port* are the fully qualified names and ports of the master host candidates. If you add master host candidates in the SAS Workload Orchestrator web interface and in metadata, you must also add them to the swo.conf file.

The value of `lbset` is incremented by one in each of the `BalancerMember` statements.

6 In the file ***SASConfig*/Levn/Web/WebServer/conf/htttpd.conf**, add this line:

```
Include conf/swo.conf
```

7 Restart SAS Web Server by running `httpdctl restart` from ***SASConfig/Levn/Web/WebServer/bin***.

Configuring SAS Workload Orchestrator Authentication

SAS Workload Orchestrator supports authorization through either HTTP Basic authentication, which uses user names and passwords, or Negotiate authentication, which uses Kerberos.

Regardless of which authentication type you use, client users on Windows must have the “Log on as batch job” user right in order for their jobs to run on the grid.

Using Basic Authentication

To use basic authentication for grid clients, do the following:

1 In the connection properties for the logical grid server definition in SAS Management Console, select an authentication domain for the grid. Do not select **<none>** as an authentication domain.

   Using SAS Management Console, open the definition for the SAS Logical Grid Server (under the SAS Application Server). Select the Grid Server definition. Under the **Connections** tab, right-click the connection entry and select **Properties**. On the **Options** tab, select or specify an authentication domain in the **Authentication Domain** field. Do not select **<none>**.

2 Specify the authentication that is needed for the SAS object spawner to connect to the grid in order to obtain information about the grid. Use one of these methods:

   - Use the credentials that are stored in metadata for the SAS Trusted User for the grid’s authentication domain. The SAS Trusted User must be specified as a grid administrator in the SAS Workload Orchestrator web interface. See “Configuring SAS Workload Orchestrator General Options” on page 41 for more information about specifying a grid administrator.
   - Use the credentials in an authinfo file for the SAS installer user. See Client Authentication Using an Authinfo File for more information.

3 If the credentials for the client users are to be stored in the authentication domain, ensure that the client users are defined in metadata.

   If the client users use credentials from an authinfo file, ensure that the file contains a user ID and password for connecting to the host and port of the grid master. Also, ensure that the file is in a location that the grid client can access. For clients that run SAS or the SAS Grid Manager Client Utility locally, use a local authinfo file. For clients that use SAS Foundation servers, use an authinfo file in the home directory of the client user. This directory must be accessible to

every machine on the grid, including the machine on which the SAS object spawner runs.

Note: On Windows, credentials must be specified as *domain\username*.

Using Negotiate Authentication on UNIX

To use Negotiate authentication for grid clients in a UNIX environment, do the following:

1. Using SAS Management Console, open the definition for the SAS Logical Grid Server (under the SAS Application Server). Select the Grid Server definition. Under the **Connections** tab, right-click the connection entry and select **Properties**. On the **Options** tab, select **<none>** in the **Authentication Domain** field.
2. Create the ***SASConfig*/Levn/ObjectSpawner/objspawn.keytab** file for the *SAS_installer_user@REALM* principal.
3. On each grid node, create the ***SASConfig*/Levn/Grid/grid.keytab** file for the HTTP/*FQDN_of_host@REALM* service principal.
4. Verify that the -SSPI option is present on the USERMODS environment variable in the ***SASConfig*/Levn/ObjectSpawner/ObjectSpawner_usermods.sh** file in order to start the SAS object spawner with the `-SSPI` option.
5. Add the -SSPI option to either the USERMODS_OPTIONS environment variable or the SSPIOPTION environment variable in the ***SASConfig*/Levn/Grid/sgmg_usermods.sh** file in order to start SAS Workload Orchestrator with the `-SSPI` option.
6. Add the -SSPI option to the _usermods.sh file for any server that will connect to the grid (such as through grid-enabled SAS/CONNECT SIGNON statements).
7. Ensure that any user accounts that are creating server definitions use IWA or SSPI to connect to the workspace server.
8. Ensure that UNIX user accounts that submit jobs to the grid have created a Kerberos credential cache, either when they log on or by using the `kinit` command.
9. Start grid clients (such as SAS, the SAS Grid Manager Client Utility, or SAS Workspace Servers) with the SSPI option.

Using Negotiate Authentication on Windows

To use Negotiate authentication for grid clients in a Windows environment, do the following:

1. Using SAS Management Console, open the definition for the SAS Logical Grid Server (under the SAS Application Server). Select the Grid Server definition. Under the **Connections** tab, right-click the connection entry and select **Properties**. On the **Options** tab, select **<none>** in the **Authentication Domain** field.
2. Use the setspn command to assign `HTTP/FQDN_of_host SPN` to the SAS Workload Orchestrator service account for every machine in the grid.
3. Verify that the -SSPI option is present on the USERMODS variable in the ***SASConfig*/Levn/ObjectSpawner/ObjectSpawner_usermods.bat** file in order to start the SAS object spawner with the `-SSPI` option.

   Note: If you selected **Use Integrated Windows Authentication** in the SAS Deployment Wizard, this value is already set.

4. Add the -SSPI option to the USERMODS variable in the ***SASConfig*/Levn/Grid/sgmg_usermods.cmd** file in order to start SAS Workload Orchestrator with the `-SSPI` option.
5. Remove and then re-install the Workload Orchestrator Service to include the added option in the Windows service parameters.

   Remove the service
   ***SASConfig*/Levn/Grid/sgmg.cmd remove**

   Re-install the service
   ***SASConfig*/Levn/Grid/sgmg.cmd install**

6. Start grid clients (such as SAS, the SAS Grid Manager Client Utility, or SAS Workspace Servers) with the SSPI option.

SAS Workload Orchestrator Invocation Options

Overview of SAS Workload Orchestrator Invocation Options

SAS Workload Orchestrator invocation options consist of options that are specific to the grid and several SAS system options that you can use to run and configure SAS Workload Orchestrator from the command line. On Windows and UNIX, when you install and configure SAS Workload Orchestrator with the SAS Deployment Wizard, a SAS Workload Orchestrator command or script file is created by default in the configuration directory (sgmg.cmd or sgmg.sh).

The configuration directory, ***SASConfig*/Levn/Grid**, contains these scripts.

Here are the valid commands for the command or script file:

- start
- stop
- restart
- status
- install (Windows only)
- remove (Windows only)

When starting SAS Workload Orchestrator, you can specify invocation options. The invocation options can be logically grouped into these categories:

- general options
- SSL options
- service options

General Options

Overview of General Options

Use the following general options for specifying an alias, FIPS compliance, the location of the logging configuration file, and use of SSPI.

- -alias
- -encryptfips
- -logconfigloc
- -sspi

Syntax Description

-alias *host_alias*
: Specifies an alias for the host in case it is known by more than one name. On UNIX, multiple aliases are allowed, and all values are used. On Windows, only one alias is allowed.

-encryptfips
: Causes SAS Workload Orchestrator to run in Federal Information Processing Standards (FIPS) compliance mode that is provided by SAS/SECURE software in its implementation of the FIPS 140-2 specification. SAS Workload Orchestrator checks each server and port to ensure that Advanced Encryption Standard (AES) is used as the encryption algorithm. If SAS Workload Orchestrator is not using AES, SAS Workload Orchestrator terminates.

  For more information, see “FIPS 140-2 Standards Compliance” in *Encryption in SAS*.

-logconfigloc *filename*

Specifies the log options for SAS Workload Orchestrator in a log configuration file. An example is `-logconfigloc "C:\SAS\Config\Lev1\Grid\logconfig.xml"` The file is an XML file that uses specific options. For more information, see "Using the SAS Logging Facility in the SAS Intelligence Platform" in *SAS Logging: Configuration and Programming Reference*.

Unless a path is specified, the spawner looks for the log config file in the current directory.

-sspi

Identifies support for the Security Support Provider Interface (SSPI) for single sign-on connections to SAS Workload Orchestrator.

For more information, see "SSPI System Option" in *SAS Intelligence Platform: Application Server Administration Guide*.

SSL Options

Overview of SSL Options

Use the following SSL options to configure SSL communications with SAS Workload Orchestrator. The required options are predefined, but are commented out in the sgmg_usermods.sh (.bat) file, which is in the directory ***SASConfig*/Levn/Grid**.

- -SSLCLIENTAUTH
- -SSLCALISTLOC
- -SSLCERTISS
- -SSLCERTLOC
- -SSLCERTSERIAL
- -SSLCERTSUBJ
- -SSLCRLCHECK
- -SSLCRLLOC
- -SSLPKCS12LOC
- -SSLPKCS12PASS
- -SSLPVTKEYLOC
- -SSLPVTKEYPASS

Syntax Description

-SSLCLIENTAUTH

specifies that SAS Workload Orchestrator should perform client authentication.

-SSLCALISTLOC=*file_path*
specifies the location of a single file that contains one or more public certificates for all the trusted certificate authorities (CA) in the trust chain. This option is used only in a UNIX environment.

-SSLCERTISS=*issuer-of-digital-certificate*
specifies the name of the issuer of the digital certificate that should be used by TLS. This option is used only in a Windows environment.

-SSLCERTLOC=*file_path*
specifies the location of a file that contains a digital certificate for the machine's public key. This is used by servers to send to clients for authentication. This option is used only in a UNIX environment.

-SSLCERTSERIAL=*serial_number*
specifies the serial number of the digital certificate that should be used by TLS. This option is used only in a Windows environment.

-SSLCERTSUBJ=*subject_name*
specifies the subject name of the digital certificate that Transport Layer Security (TLS) should use. This option is used only in a Windows environment.

-SSLCRLCHECK
specifies that Certificate Revocation Lists (CRLs) are checked when digital certificates are validated.

-SSLCRLLOC=*file_path*
specifies the location of a file that contains a Certificate Revocation List (CRL). This option is used only in a UNIX environment.

-SSLPKCS12LOC=*file_path*
specifies the location of the PKCS #12 DER encoding package file that contains the certificate and the private key. This option is used only in a UNIX environment.

-SSLPKCS12PASS=*password*
specifies the password that TLS requires in order to decrypt the PKCS #12 DER encoding package file. The PKCS #12 DER encoding package is stored in the file that is specified by the SSLPKCS12LOC= option. This option is used only in a UNIX environment.

-SSLPVTKEYLOC=*file_path*
specifies the location of the file that contains the private key that corresponds to the digital certificate that was specified by using the -SSLCERTLOC= option. This option is used only in a UNIX environment.

-SSLPVTKEYPASS=*password*
specifies the password that TLS requires in order to decrypt the private key. The private key is stored in the file that is specified by the SSLPVTKEYLOC= option. This option is used only in a UNIX environment.

SASGRID Script Variable Options

Starting with 9.4M6, use the following variables to set additional properties that are used by the sasgrid script:

SASUSERCONNECTOPTS
: specifies additional options on the SAS command line when the SAS/CONNECT server is started.

SASUSERBATCHOPTS
: specifies additional options on the SAS command line when SAS is processing in batch mode.

SASWARNINGSNOERROR
: if set to anything, changes a return code of 1 (indicating SAS completed with warnings) to a return code of 0. This change makes jobs appear to complete without warnings.

SASSTAGECMDOPTS
: specifies additional options on the command that is used to stage files.

SASREDIRSASOPTS
: specifies options on the redirection program for SAS.

SASREDIRCMDOPTS
: specifies options on the redirection program for commands.

These variables are located in the following files:

For UNIX	`SASConfig/Levn/SASAppServerContext/GridServer/grid_usermods.cfg`
For Windows	`SASConfig\Levn\SASAppServerContext\GridServer\grid_usermods.cfg`

Service Options

Overview of Service Options

Use the following service options to create, modify, and delete SAS Workload Orchestrator service definitions. These options are used only on Windows.

- -install
- -name
- -servdir
- -servuser
- -servpass
- -installdependencies
- -deinstall

For more information, see "Update a Windows Object Spawner Service" in *SAS Intelligence Platform: Application Server Administration Guide*.

Syntax Description

-install | -i <-name *name*> <-servdir *directory*> <-servuser *user ID*> <-servpass *password*>

instructs SAS Workload Orchestrator to install as a Windows service. This option can be abbreviated as -i. When asked to install as a service, SAS Workload Orchestrator records all options that are specified at installation time in the registry under the following key: `"SYSTEM\CurrentControlSet\Services\service-name\Parameters"`. You can also specify options in the start-up parameters when you manually start the spawner service from the Microsoft Windows Services snap-in (services.msc).

-name *name*

specifies a Windows service name to use when installing SAS Workload Orchestrator as a service. Use with -install. The default value is **`SAS Workload Orchestrator`**.

If you specify a service name that contains embedded blank spaces, then you must enclose the name in quotation marks (" ").

Note: If you install more than one instance of SAS Workload Orchestrator as a service on the same machine, then you must use the *-name* option to give each instance of the service a unique name.

-servdir *directory*

specifies the directory in which to run the Windows service. Use with -install. By default, the directory is ***`installation-directory`***`\Config\Lev1\Grid`.

-servuser | -su *user ID*

specifies a user name that the Windows service runs under when you also specify the *-install* option. Use with -install. This option can be abbreviated as -su.

-servpass | -sp *password*

specifies a password for the user name that is specified in the *-servUser* option. Use with -install. This option can be abbreviated as -sp.

-installdependencies | -idep *service-1*<;*service-2*><;...>

specifies the Windows services that must be started before the SAS Workload Orchestrator service starts. This option can be abbreviated as -idep.

The *service* value is the name of the dependent service that is displayed in the Microsoft Windows Services snap-in (services.msc).

-deinstall | -di –name *name*

instructs SAS Workload Orchestrator to uninstall as a Windows service. This option can be abbreviated as -di.

The *name* value is the SAS Workload Orchestrator service name that is displayed in the Microsoft Windows Services snap-in (services.msc).

Configure HTTPS Manually for SAS Workload Orchestrator

Enable HTTPS for SAS Workload Orchestrator

Complete the following steps:

1. In the SAS Workload Orchestrator web interface, select **Configuration** in the left navigation area.
2. On the Configuration page, select the **General** tab.
3. On the **General** tab, turn off the **SSL Disable** switch.

   For details about this tab, see "Configuring SAS Workload Orchestrator General Options" on page 41.
4. Click to save the configuration.
5. Stop the Grid daemon on each grid node. Issue one of the following commands:
   - `SASConfig/Levn/Grid/sgmg.sh stop` on each host.
   - `SASConfig/Levn/Grid/gridStop.py`.
6. Locate and edit ***SASConfig*/Levn/Grid/sgmg_usermods.sh**.
7. In the USERMODS_OPTIONS statement, add the following options:
   - SSLCALISTLOC specifies the location of the trust store.
   - SSLCERTLOC specifies a certificate that is valid for all of the grid nodes.
   - SSLPVTKEYLOC specifies the certificate's key.
   - SSLPVTKEYPASS specifies the password for the certificate's key. If the certificate's key is password protected, this option must be provided.

   See the following example:

   ```
   USERMODS_OPTIONS='-SSLCALISTLOC "SASHome/
   SASSecurityCertificateFramework/1.1/cacert/trustedcerts.pem" -SSLCERTLOC
   "directory_path/certificate.crt" -SSLPVTKEYLOC "directory_path/certkey.key"
   ```

   Where:
   - *SASHome* is the SAS installation directory.
   - *directory_path* is the directory path to that file in your environment.

   For more information about these options, see "SSL Options" on page 33.
8. Open SAS Management Console by selecting **Start** ⇨ **Programs** ⇨ **SAS** ⇨ **SAS Management Console**.

9 In the SAS Management Console navigation tree, expand **Environment Management** ⇨ **Server Manager** ⇨ **SASApp** ⇨ **SASApp - Logical Grid Server** and then select **SASApp - Grid Server**.

10 View the **SASApp - Grid Server Properties** page. Select the **Options** tab.

11 In the **Grid Options** field, change `noSSL=1` to `noSSL=0`.

For details about this option, see "Supported Job Options – SAS Workload Orchestrator" on page 305.

12 In the SAS Management Console navigation tree, expand **Application Management** ⇨ **Configuration Manager** and then select **Workload Orchestrator**.

13 On the **Workload Orchestrator Properties** page, select the **Internal Connection** tab.

14 From the **Communication Protocol** list, select **HTTPS**.

15 To complete enabling the HTTPS configuration, restart all SAS services.

For information about restarting SAS services, see "Overview of Server Operation" in *SAS Intelligence Platform: System Administration Guide*.

Disable HTTPS for SAS Workload Orchestrator

Complete the following steps:

1 In the SAS Workload Orchestrator web interface, select **Configuration** in the left navigation area.

2 On the Configuration page, select the **General** tab.

3 On the **General** tab, turn on the **SSL Disable** switch.

4 Click ⊟ to save the configuration.

5 Stop the Grid daemon on each grid node. Issue one of the following commands:

- *SASConfig*`/Levn/Grid/sgmg.sh stop` on each host.
- *SASConfig*`/Levn/Grid/gridStop.py`.

6 Locate and edit ***SASConfig*/Levn/Grid/sgmg_usermods.sh**.

7 In the USERMODS_OPTIONS statement, remove the following options:

- SSLCALISTLOC
- SSLCERTLOC
- SSLPVTKEYLOC
- SSLPVTKEYPASS

8 Open SAS Management Console by selecting **Start** ⇨ **Programs** ⇨ **SAS** ⇨ **SAS Management Console**.

9 In the SAS Management Console navigation tree, expand **Environment Management** ⇨ **Server Manager** ⇨ **SASApp** ⇨ **SASApp - Logical Grid Server** and then select **SASApp - Grid Server**.

10 View the **SASApp - Grid Server Properties** page. Select the **Options** tab.

11 In the **Grid Options** field, change `noSSL=0` to `noSSL=1`.

12 In the SAS Management Console navigation tree, expand **Application Management** ⇨ **Configuration Manager** and then select **Workload Orchestrator**.

13 On the **Workload Orchestrator Properties** page, select the **Internal Connection** tab.

14 From the **Communication Protocol** list, select **HTTP**.

15 To complete disabling the HTTPS configuration, restart all SAS services.

For information about restarting SAS services, see "Overview of Server Operation" in *SAS Intelligence Platform: System Administration Guide*.

Overview of SAS Workload Orchestrator Grid Management

Most organizations that use SAS consist of a variety of user categories. Each user category has its own needs and expectations. For example, your organization might have these users:

SAS Enterprise Guide and SAS Add-In for Microsoft Office users
: These users are usually running interactive programs and expect immediate results.

SAS Enterprise Miner users
: These users might be using multiple machines to train models.

SAS Web Report Studio users
: These users might be scheduling reports to run at a specified time.

SAS Risk Dimensions users
: These users might be running jobs at night.

Some users in your environment might be running jobs that have a high priority. Other users might be running jobs that require a large number of computing resources. A SAS grid environment must be able to account for all of these different needs, priorities, and workloads.

In order to manage this type of environment, you must be able to control when and where jobs can run on the grid. You can manage grid resources using these strategies in SAS Workload Orchestrator:

- Queues. They enable you to control when jobs can run and what computing resources are available to the jobs that are submitted to the queue. You can create queues based on factors such as which hosts to use or what is the job priority. When you submit a job to a particular queue, the queue settings

determine when the job runs and what priority the job has with respect to other jobs that have been submitted to the grid. You can also specify computing resources that must be available on a host in order to process a job from the queue.

- Host types. They enable you to specify job limits for particular host types and to control how many jobs can run concurrently on machines on the grid. This enables you to tune the load that each machine on the grid can accept. For example, you can assign a higher job limit to the host type for higher-capacity machines. Such machines can process many jobs concurrently.
- Tags. They enable you to specify where jobs are run on the grid by specifying tags on hosts and using matching tags on jobs. The tags are specified on machines on the grid to indicate what type of job each machine should run. When you submit jobs to the grid, you can specify tags to identify which machines should be used to process the job. You can also specify required tags on queues to limit which hosts are permitted to run jobs from a queue.

Accessing the SAS Workload Orchestrator Web Interface

The SAS Workload Orchestrator web interface enables you to monitor jobs, queues, hosts, services, and logs in a SAS Grid Manager environment. You can also configure hosts, queues, services, user groups, and user resources.

To access the SAS Workload Orchestrator web interface, open a web browser to this address:

`http://master_host:master_port`

The *master_host* and *master_port* values are specified during installation. See Step 6 on page 24 in "Installing SAS Grid Manager."

By default, *master_host* is set to the grid control server host.

You can also view the application using a theme by including the theme in the address:

`server:port/sasgrid/index.html?sap-ui-theme=theme_name`

Here are the available default themes:

sas_corporate
: This is the default theme.

sas_inspire
: This theme consists of vibrant and cohesive colors that shift the emphasis from the application to the content.

sas_hcb
: This theme has a dark background with high-contrast foreground elements to meet the needs of users with low vision.

Configuring SAS Workload Orchestrator General Options

Follow these steps to specify general settings for a SAS Workload Orchestrator configuration:

1. In the SAS Workload Orchestrator web interface, select **Configuration** in the left navigation area.
2. On the Configuration page, select the **General** tab.
3. Provide a name and description for the grid configuration, and then specify values in these fields:

   License file
   Specifies the path and filename for the SAS Workload Orchestrator license file. This value is already specified by SAS Deployment Wizard. This field is required.

   GUI directory
   Specifies the path to SAS Workload Orchestrator web interface files. This value is already specified by SAS Deployment Wizard. This field is required.

   GUI zip directory
   Specifies the path to the directory that contains the ZIP file for SAS Workload Orchestrator web interface files. This value is already specified by SAS Deployment Wizard. Do not change this value. This field is required.

   Shared directory
   Specifies the path to a directory that is shared by all grid hosts. This value is already specified by SAS Deployment Wizard. This field is required.

   Daemon port
   Specifies the port on which the daemon on each grid host listens. This value is already specified by SAS Deployment Wizard. This field is required. If you change this value, you must restart all daemons.

   SSL disable
   Specifies whether SSL authentication is enabled or disabled. By default, SSL is disabled. If you change this value, you must restart all daemons.

   Authorization disabled
   Specifies whether authorization is disabled for submitting jobs to the grid. By default, authorization is enabled, so jobs run on the grid using the credentials of the submitting user. If authorization is disabled, all jobs run on the grid as the SAS Workload Orchestrator process user. If you change this value, you must restart all daemons.

   Case sensitive user names
   Specifies whether case is considered when checking user names.

   Round robin
   Specifies the amount of time to wait after a job is sent to a host before another job can be sent to the same host. The default value is 0. Specifying a

value other than 0 causes jobs to be distributed to more hosts when a large number of jobs are submitted in a short period of time. This field is required.

Host update
Specifies the amount of time to wait between host and job updates that are sent to the master host. The default value is 15 seconds. This field is required.

Host inactive
Specifies the amount of time to wait before a host is considered inactive. The default value is 60 seconds (1 minute).This field is required.

Host unknown
Specifies the amount of time that a host can remain in an inactive state before it is purged from the grid. The default value is 900 seconds (15 minutes). This field is required.

Job change command
Specifies a script file that is called when a job is submitted.

Job purge
Specifies the amount of time that a completed job remains in memory before it is purged from the master. The default value is 3600 seconds (one hour). This field is required.

Service wait
Specifies the amount of time to wait for a service to respond to a status request. The default value is 15 seconds. This field is required.

Administrators
Specifies the users who are administrators of the grid. This field is required.

Max Standard Out Lines
Specifies the maximum number of log lines to include in standard output. If the number of log lines received exceeds the limit, older lines are discarded. This field is required. This field is available only if you have installed 9.4M8 or the hot fix for SAS releases 9.4M6 or 9.4M7.

Job update (seconds)
Specifies the amount of time that a daemon waits between updating the master with job status information. The default value is 15. This field is required.

This field is available on the **General** tab only if you have installed SAS release 9.4M8.

Master update (seconds)
Specifies the amount of time that a daemon waits between updates to the master state. The default value is 15. This field is required.

This field is available on the **General** tab only if you have installed SAS release 9.4M8.

Scheduler update (seconds)
Specifies the amount of time that a daemon waits between checking pending jobs. The default value is 15. This field is required.

This field is available on the **General** tab only if you have installed SAS release 9.4M8.

Service update (seconds)
Specifies the amount of time that a daemon waits between updating the master with high-availability service status information. The default value is 15. This field is required.

This field is available on the **General** tab only if you have installed SAS release 9.4M8.

Exporting and Importing Configurations

The SAS Workload Orchestrator web interface works only with the active configuration. If you change whether to authenticate users or the values of the **Daemon port** or **SSL disable** fields, you must restart all daemons. All other changes that you make in the configuration are applied and take effect immediately, without having to load the configuration. SAS Workload Orchestrator enables you to export all the configuration settings. You can use the exported file as a backup of the configuration, or you can save sets of configuration settings that are used in special circumstances.

To export a configuration, select in the toolbar at the top right of the Configuration page. The configuration is saved to your local machine in a JSON file named `SGMGCONFIG.`*`yyyy-mm-dd`*`.cfg`.

To import a configuration, select from the toolbar on the Configuration page. The settings in the SAS Workload Orchestrator web interface are replaced with the settings from the imported file (which must be a JSON file in the same format as the exported file).

Configuring Queues

Understanding How Queues Work

In SAS Grid Manager, job scheduling and distribution is based around queues. When a job is submitted for processing on the grid, it is placed in a queue and is held until resources are available for the job. All jobs submitted to the same queue share the same scheduling and control policy. You can define multiple queues for different priorities and classes of jobs. For example, you can define a queue for routine jobs, a queue for high-priority jobs, and a queue that prioritizes jobs that run overnight. Parameters on queue definitions enable you to control the hosts on which the jobs are processed and to specify resources that must be available.

A default queue is always present. If one is not available, it is automatically created.

SAS Grid Manager chooses which job to process based on the priority value of the queue to which the job has been sent. Priority values are relative. For example, an organization could have some jobs in a high-priority queue that has a priority value of 40, and others are in a medium-priority queue with a priority value of 20, and still others are in a low-priority queue with a priority value of 10. If these jobs ran on the same group of hosts, the jobs in the high-priority queue would be scheduled first,

followed by the medium-priority jobs, and then the low-priority jobs. After priority is considered, jobs are processed using a first-in, first-out order.

To prevent jobs from competing for the same hosts, you can also assign different hosts or host types to each queue. By assigning one group of hosts for high-priority jobs and a different group for medium- and low-priority jobs, you can ensure that low-priority jobs are not held, waiting for higher-priority jobs to process.

For each queue, you can also specify a list of other queues that can be preempted. If a job is sent to a queue that cannot find an available host to run the job, the queue checks its configuration to find queues that it can preempt. If the original queue can preempt another queue, it suspends the job on the preempted queue and sends the job to the preempted queue for processing. After the job finishes, the suspended job is resumed on the preempted queue.

Defining a High-Priority Queue

This example describes the basic process for creating a new queue and setting its priority. Setting other queue parameters are discussed in other topics. Follow these steps to create a queue:

1. In SAS Workload Orchestrator, select **Configuration** in the left navigation area.
2. On the Configuration page, select the **Queues** tab
3. Select **New queue** and provide a name.
4. Specify the priority for the queue in the **Priority** field. The default queue uses a priority value of 10. If you are creating a high-priority queue, you can specify any value greater than 10 to ensure that jobs from this queue are processed first. However, you should choose a value that allows for queues with intermediate priority values.

   In this example, a value of 40 is used for the **Priority** field.
5. When you have made all configuration changes, select to save all configuration changes and reconfigure the grid.

Automatically Requeue Restarted Jobs

You can specify that jobs are automatically requeued if they are restarted. Follow these steps to requeue restarted jobs:

1. In SAS Workload Orchestrator, select the **Queues** tab from the **Configuration** page and expand a queue definition.
2. Select the **Restart jobs** option.
3. When you have made all configuration changes, click to save all configuration changes and reconfigure the grid.

Specifying Queue Job Limits

You can limit the number of jobs that are running in a queue in order to prevent the queue from being overloaded or monopolized by a single user. Follow these steps to specify job limits:

1. In SAS Workload Orchestrator, select the **Queues** tab from the **Configuration** page and expand a queue definition.
2. In the **Maximum Jobs** field, specify the maximum number of jobs from the queue that can be running at one time.
3. In the **Maximum jobs per user** field, specify the maximum number of jobs from each user that can be running from the queue at one time.
4. In the **Maximum jobs per host** field, specify the maximum number of jobs that can be running on each host from the queue at one time. Note that the host specification also includes a max jobs parameter, which is the total number of jobs the host processes from all queues.
5. When you have made all configuration changes, select to save all configuration changes and reconfigure the grid.

Controlling Where Jobs from a Queue Are Processed

After a job is sent to a queue, several parameters are evaluated to determine which host will be used to process the job. The queue uses the values that are specified for all these parameters:

Specified hosts
: You can specify specific hosts, host groups, and host types in a queue definition using the **Host names** and **Host types** fields. If you specify hosts for the queue, the queue can select only from those hosts when choosing where to process a job.

Tags
: A tag is a label that can identify specific needs or capabilities. You can specify tags for host definitions and specify required tags for queue definitions in the **Required tags** field. If a queue definition includes one or more required tags, only hosts that have the matching tags are able to process jobs from that queue.

Consumed resources
: A consumed resource is a host computing resource that must be available in a specified quantity in order for a job to be processed on the host, and the same specified quantity must be available on the host each time a job is processed. For example, if you specify 1024 MB of memory as a consumed resource, the queue searches for a host that has 1024 MB of memory available to process the job. When the queue sends the job to the host, it decrements the amount of memory that it knows to be available on that host by 1024 MB. In general, a

queue definition specifies either required resources or consumed resources, but not both.

Required resources
: A required resource is a host computing resource that must be available in a specified quantity in order for a job to be processed on the host. For example, if you specified a value of four cores as a required resource, only hosts that have at least four cores can process jobs from the queue. In general, a queue definition specifies either required resources or consumed resources, but not both.

Compare order
: If, after evaluating the specified hosts, tags, and required resources, the queue identifies multiple hosts that are eligible to process the job, it will choose the host according to a resource comparison order (compare order). For example, for the compare order of utilization followed by used memory, the utilization resource on each host would be compared first, and the host with the lowest utilization would be chosen. If more than one host has the same level of utilization, the amount of used memory on each host would be compared next. Each resource type value specifies whether smaller or larger values are preferable. The compare order applies to all queues.

Note: User-defined resources are not evaluated in the compare order.

Follow these steps to specify these parameters:

1. In SAS Workload Orchestrator, select the **Queues** tab from the **Configuration** page and expand a queue definition.

2. Specify specific hosts or host groups that can process jobs from the queue. Enter the host names or host group names in the **Host names** field. When specifying multiple hosts, press **Enter** after each host.

   Leave the field blank if all hosts can process jobs from the queue.

   To specify that a host should not process jobs from the queue, specify `!host_name`.

3. Specify tags that must be matched with tags in host type definitions in the **Required tags** field. When specifying multiple tags, press **Enter** after each tag.

4. Specify any consumed resources for the queue. Select the **Queues** tab and expand the queue definition. In the **Consumed Resource** area, click the **Select a item** field and select the resource that you want to specify. After you select the resource, click **+** and specify a value for the selected resource in the field that appears. The value that you specify is both the amount of the resource that must be available and the amount of the resource that is decremented by each job. These resources are available by default:

   cores
   : Number of cores available

   memory
   : Total system memory (specify in MB)

   swap
   : Amount of swap memory on UNIX or temporary memory on Windows (specify in MB)

temp
Amount of temporary disk storage space (specify in MB). On UNIX, this value is the amount of storage space available in the `/tmp` path. On Windows, this value is the amount of storage space available on the disk containing %TEMP% as evaluated by the SAS Workload Orchestrator.

In addition to the default resources, you can select user-defined resources. See "Defining User-Defined Resources" on page 60 for information about defining your own resources.

5 Specify any required resources for the queue. Select the **Queues** tab and expand the queue definition. In the **Required Resource** area, click the **Select a item** field and select the resource that you want to specify. After you select the resource, click + and specify a value for the selected resource. The value that you specify is the amount of the resource that must be available on the host. The resources that are available by default are the same ones that are available for **Consumed Resources**.

   In addition to the default resources, you can select user-defined resources. See "Defining User-Defined Resources" on page 60 for information about defining your own resources.

6 When you have made all configuration changes, select to save all configuration changes and reconfigure the grid.

Specifying Queue Users and Administrators

You can specify that only certain users and groups of users can submit jobs to a queue. You can also specify administrators for the queue. Follow these steps to specify queue users and administrators:

1 In SAS Workload Orchestrator, select the **Queues** tab from the **Configuration** page and expand a queue definition.

2 In the **Users** field, specify the users and user groups that are allowed to submit jobs to the queue. If you leave this field blank, all users can submit jobs to the queue.

3 In the **Administrators** field, specify the users that can serve as administrators for the queue. Queue administrators can view and manage all jobs submitted to the queue, not just their own jobs. They can also open or close the queue and activate or inactivate the queue.

4 When you have made all configuration changes, select to save all configuration changes and reconfigure the grid.

Supporting Kerberos Constrained Delegation in Windows

If you are using Windows and have activated Windows Defender Credential Guard, you might want to activate Kerberos constrained delegation. When using constrained delegation, SAS Grid Manager must use a special handshake to provide credentials to its spawned processes. This handshake is provided for almost all types of grid processing. However, the handshake is not provided if you are using the SAS Grid Manager Client Utility to run a command that includes the `-GRIDRUNCMD` argument in the `SASGSUB` statement. If you are submitting jobs using this command and argument that run SAS processes that understand the handshake, you must provide support for constrained delegation. Here are two ways to apply constrained delegation:

- To provide support for only one job, specify the `-GRIDJOBOPTS exchangeCreds=1` option in the `SASGSUB` statement for the job.
- To provide support for all jobs submitted to a queue, specify support for constrained delegation on the queue to which jobs are submitted. Select the **SAS credential exchange** on a queue definition if the SAS credentials can be passed to the spawned processes. This enables SAS processes to access Windows services such as the SQL server.

If you are using Kerberos constrained delegation in Windows and you do not specify support using either the `exchangeCreds` option or the **SAS credential exchange** option, SAS processes can access only the local file system. The processes cannot access remote services.

If you are using constrained delegation and want to access UNC paths:

Note: You must be a domain administrator.

1. Start the Windows administrative program Active Directory User and Computers.
2. In the **Computers** folder, right-click a SAS Workload Orchestrator host and select **Properties** from the pop-up menu.
3. In the Properties dialog box, select the **Delegation** tab.
4. Select the radio buttons **Trust this computer for delegation to specific services only** and **Use any authentication protocol**.
5. Click **Add** and add an entry in the services table for each CFIS service to which the SAS process can connect.
6. Click **OK** to save the changes.

If you are using constrained delegation and want to access a Kerberos protected service such as Microsoft SQL server::

Note: You must be a domain administrator.

1. Start Active Directory User and Computers.

2 Locate the entry for the domain account under which SAS Workload Orchestrator runs. To locate the entry for account, use the command `setspn -L` *`domain-accout-name`*. Right-click the entry for the account and select **Properties** from the pop-up menu.

3 In the Properties dialog box, select the **Delegation** tab.

4 Select the radio buttons **Trust this computer for delegation to specific services only** and **Use any authentication protocol**.

5 Click **Add** and add an entry in the services table for each service to which the SAS process can connect.

6 Click **OK** to save the changes.

The Job Information page indicates whether credential exchange was used in a job. The **SAS credential exchange (request)** field specifies whether a job requested credentials, and the **SAS credential exchange (process)** field specifies whether the job used credentials passed to the job.

Configuring Hosts

Overview

Hosts are machines that process jobs that are sent to the grid and run services. One host is designated as the master. The master controls operations on the grid, routing jobs from queues to the appropriate hosts for processing. Because the master is vital to the operation of the grid, it can be configured as highly available. Multiple hosts can be designated as master candidates, although only one host acts as the master at a time. During operation, the master candidates periodically ping the master host. If the master host does not respond, the candidates ping the first host in the candidate list. If that host does not respond, they ping the next host in the list and continue going through the list until a candidate host responds. The first host to respond becomes the new master host.

You can define a number of parameters to control processing on hosts, such as the number of jobs that can be running on the host at one time. You can also specify tags in host definitions to control where jobs are processed. When the master host evaluates which host to use to process a job from a queue, it checks the tag values on the queue and host, and sends the job to the host only if the tags match.

You can associate hosts or host types with queues, ensuring that particular hosts are used to process particular jobs. You can use host groups and host types to group hosts in order to make host management easier. You can also associate hosts or host types with services.

Adding a New Host to the Grid

Hosts are added to the grid by matching them to an existing host type. If the new host matches the host name pattern, IP address pattern, or IP address range of an existing host type in SAS Workload Orchestrator, the new host is automatically added to the matching host type.

Follow these steps to add a new host to the grid:

1. If you are using a shared configuration, run this command on the new host machine:

   ```
   SASConfig/Levn/Grid/sgmg.sh start
   ```

2. If you are not using a shared configuration, run the SAS Deployment Wizard on the new host machine. Configure the new host machine as a SAS Grid Manager Grid Node.

3. If the new host matches an existing host type, the host is automatically added to the matching host type. If the new host does not match with an existing host type, it is added to the default host type.

   If you want to create a new host type for hosts that do not match an existing host type, see “Defining a New Host Type” on page 50 for information.

Defining a Host Group

Host groups enable you to specify a group name, rather than individual host names, when defining host types, queues, or services.

Follow these steps to define a host group:

1. In the SAS Workload Orchestrator web interface, select **Configuration** in the left navigation area.

2. On the Configuration page, select the **Host Groups** tab

3. Select **New host group** and provide a name and description.

4. Specify the hosts in the **Members** field. Press Enter after each entry.

5. When you have made all configuration changes, select ⊟ to save all configuration changes and reconfigure the grid.

Defining a New Host Type

You can use host definitions to specify grid processing parameters for single hosts or groups of hosts. These parameters enable you to specify the number of jobs that

the hosts can process at a time, as well as metric thresholds for scheduling and running jobs.

Follow these steps to define a new host type:

1. In the SAS Workload Orchestrator web interface, select **Configuration** in the left navigation area.
2. On the Configuration page, select the **Host Types** tab
3. Select **New host type** and provide a name and description.
4. Specify whether the host type definition is for a **Master** or **Server** role. A master host controls the distribution of jobs to the grid in addition to processing grid jobs. A server runs jobs that are sent to the grid for processing.

   You should identify more than one host as a master for high availability, although only one host can serve as the master at a time.
5. Specify the hosts that are included in the host type.

   If you are defining a master host type, specify a fully qualified host name.

   If you are defining a server host type, specify the hosts in the **Host Identification** fields. You must specify hosts using at least one of these fields:

   Host names
   Specify individual short host names.

   Host name patterns
   Specify regular expressions to match against host short names or aliases.

   IP patterns
   Specify regular expressions to match against host's IP addresses.

   IP ranges
   Specify Classless Inter-Domain Routing (CIDR) range expressions to match against IPV4 or IPV6 addresses or CIDR subnets.

   If you make multiple entries, press Enter after each entry.
6. If you are specifying a master host type and the URL `http(s)://` *`master_host`*`:`*`port`* is not valid from a client machine, you must specify the external connection information for the host. Select the **Use external connection information** check box and specify the connection protocol (HTTP or HTTPS), the external host name, the external port, and the service. The external connection is specified in the form [*protocol*]://[*hostname*]:[*port*][*service*].

   This field is visible only if you are specifying a master host type.
7. If each master candidate has a different external URL, you must create a separate host type entry for each master candidate.
8. In the **Max jobs allowed** field, specify the maximum number of jobs that are allowed to be processed at a time on any machine of this host type. Use these values when specifying the maximum number of jobs allowed:

   Negative value
   specifies that the maximum number of jobs allowed is the negative of the specified value multiplied by the number of cores. For example, if you specify a value of -4 for the **Max jobs allowed** field, and a host contains 8 cores, the host can run a maximum of 32 jobs at a time (8*4).

Positive value
: specifies that the host can run a maximum of the specified number of jobs at one time.

0 (zero)
: specifies that the host cannot run any jobs. This value is useful if you use a time-based override. You can specify that the host cannot process any jobs during the default period, and then specify a maximum value during the override period.

9 Use the **Tags** field to specify appropriate tags for the hosts. Tags are labels that can identify specific needs or capabilities. You can specify tags in host definitions and specify required tags in queue definitions. If a queue definition includes one or more required tags, only hosts that have the matching tags are able to process jobs from that queue.

10 When you have made all configuration changes, select to save all configuration changes and reconfigure the grid.

11 If you added new master host candidates, you must also update the master host candidate list in metadata. See "Updating Master Hosts in Metadata" on page 55.

If you added new master host candidates and you enabled master host failover using the SAS Web Server, you must also add any new master host candidates to the `SASConfig/Levn/Web/WebServer/conf/swo.conf` file. See "Enabling Master Host Failover for Job Flow Scheduler" on page 26 for more information.

Specifying Host Schedule Thresholds

In order to prevent new jobs from overloading a host, you can specify schedule thresholds for the host type. A schedule threshold specifies a metric limit or range that is checked before new jobs are run on a host. If the metric exceeds the threshold, new jobs are not run on the host.

Follow these steps to specify a host schedule threshold:

1 In the SAS Workload Orchestrator web interface, select the **Host Types** tab from the **Configuration** page and expand a host type definition.

2 In the **Schedule Thresholds** area, use the **Select an item** field to select a metric to use for the schedule threshold. These metrics are available by default:

ioRate
: the number of bytes that a job can read and write in a second

pgRate
: the number of pages that a job can read and write in a second

runQueue15s
: the CPU run queue length averaged over the course of 15 seconds

runQueue1m
: the CPU run queue length averaged over the course of one minute

runQueue15m
: the CPU run queue length averaged over the course of 15 minutes

usedMemory
the total amount of memory used, in MB

usedSwap
the amount of swap memory used, in MB

usedTemp
the amount of temporary memory used, in MB

utilization
the percentage of CPU time used

In addition to the default threshold metrics, you can also select any host dynamic resources that you define.

3 Click + beside the metric that you selected. A set of fields appears that enable you to select operators and values to define the threshold range.

4 Repeat steps 2 and 3 to define additional thresholds.

5 When you have made all configuration changes, select ⊟ to save all configuration changes and reconfigure the grid.

Specifying Host Suspend Thresholds

In order to prevent running jobs from overloading a host, you can specify suspend thresholds for the host type. A suspend threshold specifies a metric limit or range that is checked while a job is running on the host. If the metric exceeds the threshold, the jobs that are running on the host are suspended. Jobs are suspended one at a time, 15 seconds apart. When the metric drops below the threshold limit, jobs are then restored one at a time.

Follow these steps to specify a host suspend threshold:

1 In the SAS Workload Orchestrator web interface, select the **Host Types** tab from the **Configuration** page and expand a host type definition.

2 In the **Suspend Thresholds** area, use the **Select an item** field to select a metric to use for the suspend threshold. The metrics that are available are the same as those used for schedule thresholds. See "Specifying Host Schedule Thresholds" on page 52 for a list of available metrics.

In addition to the default threshold metrics, you can also select any host dynamic resources that you define.

3 Click + beside the metric that you selected. A set of fields appears that enable you to select operators and values to define the threshold range.

4 Repeat steps 2 and 3 to define additional thresholds.

5 When you have made all configuration changes, select ⊟ to save all configuration changes and reconfigure the grid.

Reordering Master Hosts

During operation, hosts send their status to the master host. If the master host does not respond, the daemons contact the first host in the candidate list. If that host does not respond, they contact the next host in the list and continue going through the list until a candidate host responds. The first master host candidate to respond becomes the new master host.

The order of the master hosts is determined by the order that the master hosts and master host types are specified in the SAS Workload Orchestrator configuration. During installation, the master host and a list of master host candidates are specified. The initial configuration includes a master host type that lists the master hosts in the order in which they were specified during installation. If you add master host types, the master hosts in the types are added to the order in which the master hosts are checked.

The first host that you define as a master host is the primary master host, the second master host defined in the first master host candidate, the third master host is the second master host candidate, and so on. If you have defined more than one master host type, the hosts in the first master host type are used before the hosts in the next master host type. Within a master host type, the hosts are used in the order in which they are listed.

For example, if you define host types of MasterType1 (which contains hostA and hostB), ServerType1 (which contains hostC and hostD), and MasterType2 (which contains hostQ and hostW), the list of master candidates is hostA, hostB, hostQ, and hostW (in that order).

You can use the SAS Workload Orchestrator web interface to change the order of master host types. Follow these steps:

1. In the SAS Workload Orchestrator web interface, select **Configuration** and select the **Host Types** tab.

   The tab lists all host type definitions. The first master host type definition that is listed is the master host. The other master host candidates are listed in descending order. However, because the tab lists master and server host types together, you must open the host type definition to determine whether it is a master or server host type.

2. To move a server higher in the order, select ▲ on the row for the host type. If you move a master host candidate to the top of the list of master hosts, it becomes the new master host.

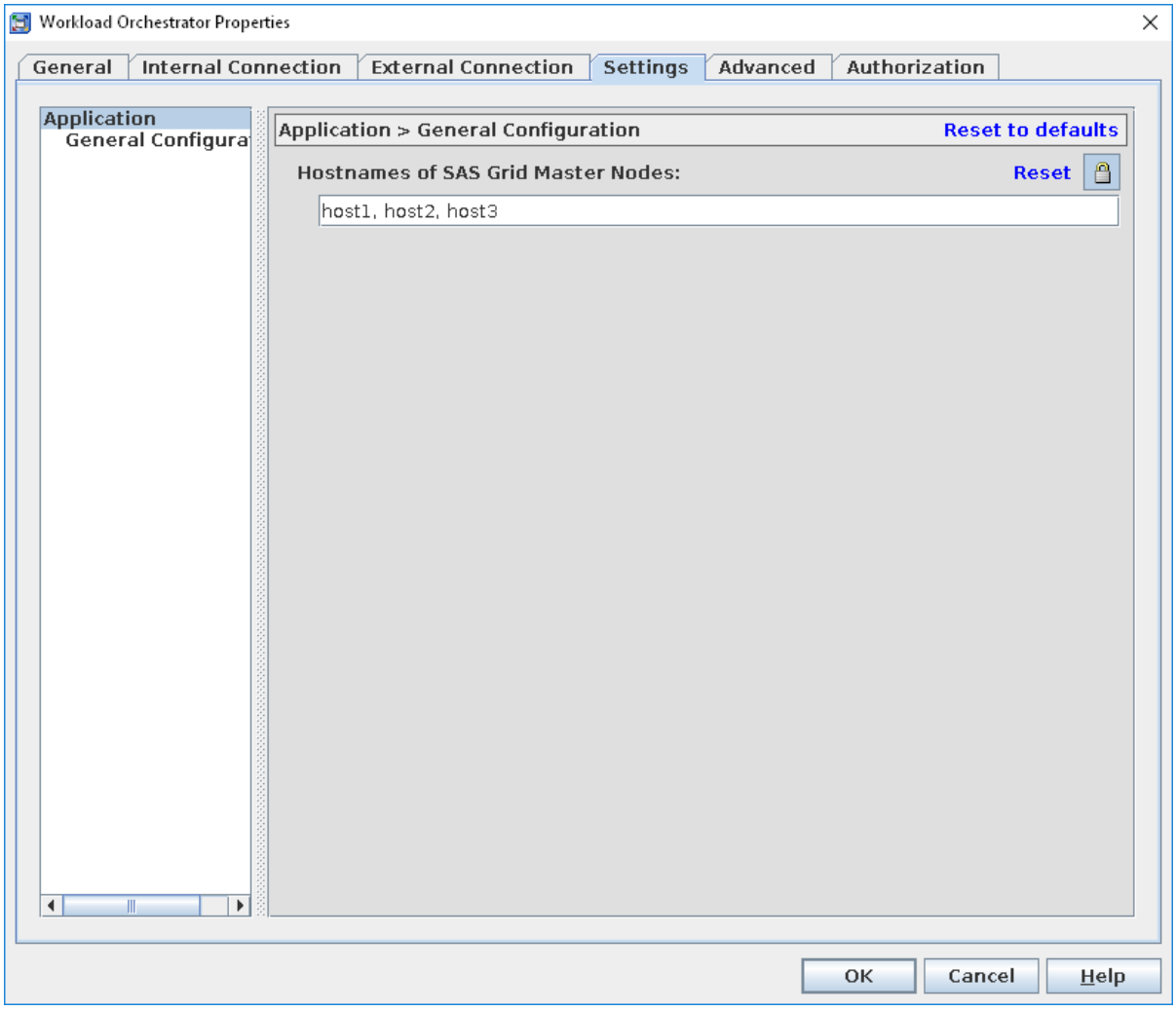
   Note: If you change the master host or the master host candidate order, you must also change the values in metadata in order for the SAS Workload Orchestrator agent plug-in to monitor the grid. See "Updating Master Hosts in Metadata" on page 55.

3. To move a server lower in the order, select ▼ on the row for the host type.

Updating Master Hosts in Metadata

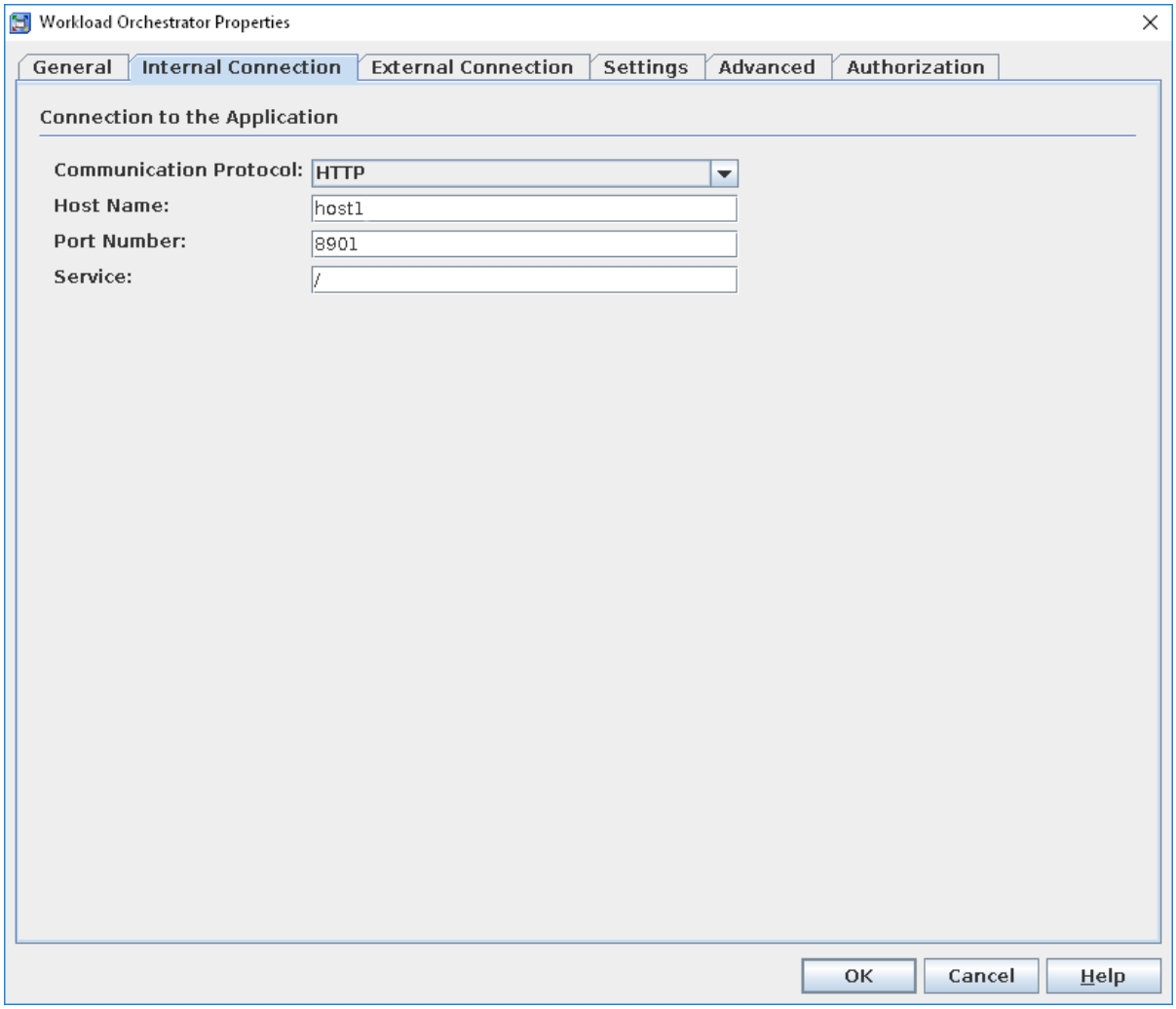
When you specify the master host and master host candidates during installation, the host names are stored in metadata. If you use the SAS Web Orchestrator web interface to change the master host or add hosts to the master host candidate list, these changes are not also made in metadata. In order for the SAS Grid Manager agent plug-in for SAS Environment Manager to monitor the grid, you must manually change the master host information in metadata. To use SAS Management Console to change the master host information in metadata, follow these steps:

1. In SAS Management Console, select the **Configuration Manager** plug-in.
2. In the list of configuration entries, right-click **Workload Orchestrator** and select **Properties**.
3. Select the **Settings** tab.
4. The field **Hostnames of SAS Grid Master Nodes** field contains a comma-separated list of all the master host candidates. The first host in the list is the master host. The other hosts in the list are master host candidates that can take over as the master host if the current master fails.



5 To add master host candidates, add comma-separated values to the list.

6 To change the master host, change the name of the host that is the first in the list. You must then select the **Internal Connection** tab and specify the same host name in the **Host Name** field. Click **OK** to save the changes.



7 In SAS Management Console, under the **Server Manager** plug-in, expand the entry for the **SAS Application Server**, and then expand the entry for the logical grid server.

8 Below the entry for the logical grid server, right-click the entry for the grid server, select **Properties**, and select the **Options** tab.

9 In the **Grid Options** field, specify the new master hosts as values for the masterList option. The list of master host candidates is enclosed in single quotation marks and separated by commas. The first host in the list is the master host. Click **OK** to save your changes.

10 If you changed the first master host, on the **Connections** tab, right-click on the entry for the grid server connection and select **Properties**.

11 Select the **Options** tab and specify the first master host in the **Grid Server Address** field.

12 If you enabled master host failover using the SAS Web Server, you must also add any new master host candidates to the `SASConfig/Levn/Web/WebServer/conf/swo.conf` file. See "Enabling Master Host Failover for Job Flow Scheduler" on page 26 for more information.

See "Using SAS Environment Manager" on page 76 for information about the SAS Workload Orchestrator agent plug-in.

Specifying Time-Based Queue or Host Type Parameters

After you specify the default settings for a queue or a host type, you can specify different settings that are used only during specified time periods. For example, you could create a queue for overnight processing that has a default priority of 0 but a priority of 40 between 10 PM and 2 AM.

If you specify multiple overrides, they are evaluated in the order in which they are listed (left to right in the SAS Workload Orchestrator web interface). You can change the override order by dragging and dropping the override tabs. If more than one override is valid at a specified time, the values are used from the first valid override.

The same concept applies for host type definition overrides.

Follow these steps to create a time override:

1. In SAS Workload Orchestrator, select **Configuration** in the left navigation area.
2. On the Configuration page, select either the **Queues** tab or the **Host Types** tab, depending on the type of override you are defining.
3. Expand the entry for a queue or host type. The parameters specified in the **Default Time Based Settings** area are used whenever a time override is not in effect.
4. Click + next to the **Default Time Based Settings** label. The Edit Configuration Time Override window appears.
5. Specify a name for the override and the time that the override is active. In the **Start** field, specify a cron expression for the time at which the override takes effect. In the **End** field, specify a cron expression for the time at which the override ends. See "Writing cron Expressions" on page 315 for more information.
6. To specify additional times when the override is active, click **Add interval** and specify starting and ending cron expressions.
7. Select **OK** after specifying all the override intervals.
8. The Configuration page for the queue or host type is displayed, with the tab for the override interval selected. Specify parameters for the queue or host type that are used during the selected interval.
9. When you have made all configuration changes, select to save all configuration changes and reconfigure the grid.

Specifying Host Comparison Order

When jobs are submitted to the grid, SAS Workload Orchestrator evaluates several criteria before determining which host the job runs on. After checking the job and queue specifications, hosts are evaluated based on the resource comparison order. For example, you could specify that the host with the largest number of cores be chosen first. You can also specify that if more than one host has the same number of cores, then the host with the largest amount of total memory is chosen.

The default host comparison order is runQueue15s followed by ioRate.

Follow these steps to specify the host comparison order:

1. In the SAS Workload Orchestrator web interface, select **Configuration** in the left navigation area.
2. On the Configuration page, select the **Compare Order** tab.
3. Select resources in the **Available Resources** list and click **+>** to move them to the **Compare Order** list. Items that are higher in the **Compare Order** list are compared before items that are lower in the list.

Configuring User Groups

User groups enable you to control access to queues by certain categories of users. Rather than having to individually specify the users that can access a queue, you can specify one or more user groups, and then manage the group membership.

To create a user group, follow these steps:

1. In SAS Workload Orchestrator, select **Configuration** in the left navigation area.
2. On the Configuration page, select the **User Groups** tab.
3. Select **New user group** and specify a name for the group.
4. Specify the user IDs or names for members of the group in the **Members** field.
5. Specify the users who have administrative access to the group in the **Administrators** field. A user group administrator can monitor and manage jobs from other users in the group.

Note: You can assign a user group administrator only if the user group is assigned to the execution queue. See “Specifying Queue Users and Administrators” on page 47.

6 When you have made all configuration changes, select ⊟ to save all configuration changes and reconfigure the grid.

Defining User-Defined Resources

A resource in SAS Workload Orchestrator is a measurement of a computing resource on the grid. These measurements are used in the selection and ordering of hosts to process a job or to determine whether to suspend an active job. You can associate resources with queue definitions (in the **Consumed Resources** or **Required Resources** fields), host type definitions (in the **Schedule Thresholds** and **Suspend Thresholds** fields), and a host comparison order. Resources can be static or dynamic.

A static resource is one that does not vary over the life of the grid or a host. Some static resources can also be consumed whenever a job is submitted. Examples of static resources are the total memory of a host, a host's disk space, or the number of licenses on a host or on the grid. The value of a static resource for the entire grid is assumed to not change until the grid configuration changes. The value of a static resource for a grid host is assumed to not change until the host restarts or reconnects to the grid.

A dynamic resource is one that constantly changes, such as a host's CPU utilization, I/O rate, and the amount of memory used. SAS Workload Orchestrator retrieves the values of dynamic resources at periodic intervals.

SAS Workload Orchestrator provides pre-defined static resources (cores, memory, swap, and temp) and dynamic resources (ioRate, pgRate, runQueue15s, runQueue1m, runQueue15m, usedMemory, usedSwap, usedTemp, and utilization). If you want to use resources other than these, you can define your own resources.

To define a user resource, follow these steps:

1 In the SAS Workload Orchestrator web interface, select **Configuration** in the left navigation area.

2 On the Configuration page, select the **User Resources** tab.

3 Select **New user resource** and specify a name for the resource.

4 In the **Type** field, select the type of resource to be defined. Here are the choices:

Global Static
A computing resource whose value is static over the entire grid. Global values are read by a host when that host becomes the master.

Global Static Consumable
A computing resource whose value is static over the entire grid and that can be consumed by jobs as they are submitted. An example of a global static resource is the total number of licenses on the grid. Global static consumable resources can be associated with queues as a required resource or a consumable resource. Global values are read by a host when that host becomes the master.

Host Dynamic
A host computing resource whose value constantly changes. Dynamic resource values are also used in the host comparison order and host type definitions (as schedule thresholds and suspend thresholds). An example of a host dynamic resource is the amount of available SAS work space. A host dynamic resource is updated whenever the value of the resource is checked.

Host Static
A computing resource whose value is static for each host. An example of a host static resource is the number of GPUs on a host. Host static resources can be associated with queues as a required resource or a consumable resource. A host static resource value is determined when a host identifies itself to the grid master.

Host Static Consumable
A computing resource whose value is static for each host and that can be consumed by jobs as they are submitted. An example of a host static resource is the number of GPUs on a host. Host static consumable resources can be associated with queues as a required resource or a consumable resource.

5 Select the **Greater is better** check box to specify that larger values of the resource are better when using the resource in the host comparison order.

6 Specify whether the value of the resource is a static value or is determined from a script.

If you choose **Static** for the **Value type** field, you must enter a numeric value for the resource in the **Value** field.

If you choose **Script** in the **Value type** field, you must enter the path to a script that returns the value of the resource in the **Script file path** field. The name of the resource is passed to the script, and the script must return to STDOUT a numeric value of the resource. See “Sample User-Defined Resource Scripts” on page 317 for sample scripts.

7 When you have made all configuration changes, select to save all configuration changes and reconfigure the grid.

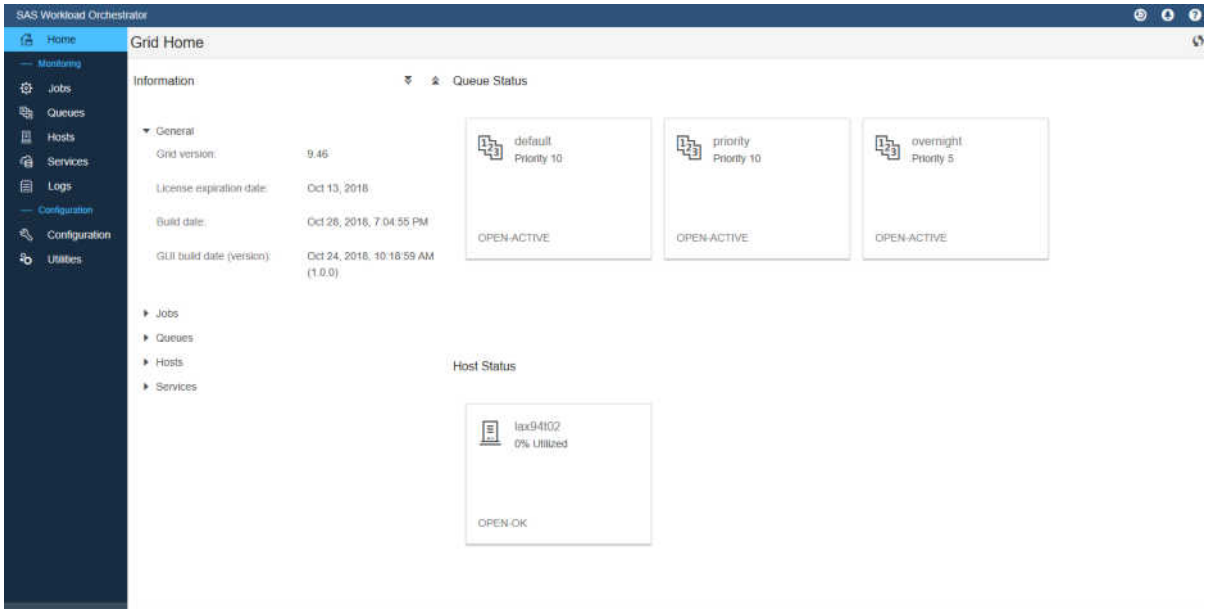
3

Managing and Monitoring a SAS Grid Manager Grid

Using the SAS Workload Orchestrator Web Interface

Using the SAS Workload Orchestrator Web Interface Home Area

The **Home** area of the SAS Workload Orchestrator web interface displays a summary of status information for the SAS Workload Orchestrator grid.



The **Information** pane on the left side of the window displays summary information about the SAS Workload Orchestrator software, the number of jobs on the grid, the number of defined queues, the number and type of grid hosts, and the defined high availability services.

The right side of the window displays tiles in the **Queue Status** and **Host Status** areas, representing the defined queues and hosts. Each tile provides a quick visual representation of the queue or host. Queue tiles display the name, priority, state, and the number of jobs that are running on the queue. Host tiles display the host name, utilization, state, and the number of jobs that are running on the host.

Use the links in the left navigation area to access these functional areas, which enable you to monitor and configure the grid:

Jobs

Monitor jobs on the grid and perform actions such as suspending, resuming, and canceling jobs. See “Monitoring and Managing Jobs” on page 65.

Queues
Monitor the operation of the queues and perform actions such as opening, closing, activating, and deactivating queues. See "Managing and Monitoring Queues" on page 68.

Hosts
Monitor the hosts on the grid, and set the hosts to be open or closed. See "Managing and Monitoring Hosts" on page 70.

Services
Monitor high availability services. See "Managing and Monitoring High Availability Services" on page 73.

Logs
View log messages from grid-related loggers. See "Viewing Log Messages" on page 74.

Configuration
Configure grid resources such as queues, host groups, services, and user groups. See "Configuring SAS Workload Orchestrator General Options" on page 41.

Utilities
Set logging levels for grid-related loggers and renew the SAS Workload Orchestrator license. See "Using the SAS Workload Orchestrator Utilities" on page 75.

Grid administrators can access all functions in the SAS Workload Orchestrator web interface.

Users who are not grid administrators can access only **Jobs**, **Queues**, and **Hosts**, and can view information only about their own jobs in these areas. Queue administrators can view information about all jobs in the queue for which they are an administrator.

Monitoring and Managing Jobs

Monitoring Jobs

Access the **Jobs** area by selecting **Jobs** from the left navigation area of the SAS Workload Orchestrator web interface.

The **Jobs** area lists all the active jobs, including those that are pending, running, or recently completed. Grid administrators can see all jobs. Non-grid administrators can see only their own jobs, jobs submitted by users in a user group for which they are a user group administrator, and jobs in queues for which they are a queue administrator. The table in the **Jobs** area identifies the job ID, the name and state of the job, the user who submitted the job, the queue that is used, the host on which the job is running, the time at which the job was submitted, and the start and end time of the job. See "Job States" on page 67 for possible values for the job state.

You can filter the list to locate the jobs that you are interested in. See "Filtering the Jobs Table" on page 66 for information.

To view detailed information about a job, click the name of the job in the **Jobs** table. Click ⁝☰ or **Close** to return to the **Jobs** table.

To perform an action on a job, select the entry for the job in the table, and then select an action. You can also select the check boxes next to multiple jobs in order to perform an action on all the selected jobs. Here are the available actions:

⏸ (**Suspend**)
Suspends the execution of the selected job. If you are an administrator, you can suspend any job for the group for which you are an administrator (for example, a user group administrator can suspend jobs only from users in their user group, and a grid administrator can suspend jobs from any grid user). If you are not an administrator, you can suspend only your own jobs.

▷ (**Resume**)
Resumes processing of a suspended job.

■ (**Cancel**)
Stops execution of the selected job. If you are an administrator, you can cancel any job for the group for which you are an administrator. If you are not an administrator, you can cancel only your own jobs.

Filtering the Jobs Table

The Jobs page enables you to filter the list of jobs in order to locate particular jobs of interest.

Enter text in the **Filter** field at the top of the page to find jobs that contain the specified string. The table displays any jobs that match the supplied string, regardless of which column contains the string.

To display a selection of jobs that match a specified set of criteria, use the **Advanced Filter** fields. You can specify multiple filter criteria. Here are the available filters:

Status
Selects the status of jobs that you want to display. Selections are ALL, ACTIVE, PENDING, RUNNING, SUSPENDED, FINISHED, and ARCHIVED. FINISHED indicates that the job has completed.

Start Job ID
Specifies the beginning of a range of job IDs to display.

Last Job ID
Specifies the end of a range of job IDs to display.

Execution Host
Specifies the name of the host on which the job ran.

Queue
Specifies the name of the queue to which the job was submitted.

User
Specifies the ID of the user who submitted the job.

After you specify the filter criteria, click **Apply**. The jobs that match the selected filter criteria are displayed in the Jobs table, and the filter fields are hidden. Click ▽ to display the filter fields.

Job States

Here are the possible values for the job state as displayed in the jobs table:

COMPLETED
: The job ran to completion and returned a zero return code.

FAILED
: The job ran to completion but returned a nonzero return code.

HOST-FAILED
: The job was killed because the execution host failed and the job was not marked as being rerunnable.

KILLED
: The job was killed by a request from the administrator, the queue administrator, or the user that submitted the job.

KILLED-LIMIT
: The job was killed because it exceeded a specified limit.

LAUNCH-ERROR
: The command for the job could not be executed.

PENDING
: The job has been submitted and is waiting in the queue for execution resources.

PENDING-SUSPENDED-ADMIN
: The job was suspended due to an administrator request while the job was in a PENDING state.

RUNNING
: The job is actively running on an execution host.

RUNNING-SUSPENDED-ADMIN
: The job was suspended due to an administrator request while the job was in the RUNNING state.

RUNNING-SUSPENDED-THRESHOLD
: The job was suspended because a suspend threshold limit was exceeded on the execution host while the job was in the RUNNING state.

RUNNING-SUSPENDED-PREEMPTION
: The job was suspended because a job from a higher priority queue preempted the job while it was in a RUNNING state.

STARTING
: The job has been assigned to a specific host and has been sent for execution, but has not yet started.

STARTING-SUSPENDED-ADMIN
: The job was suspended due to an administrator request while the job was in the STARTING state.

STARTING-SUSPENDED-THRESHOLD
: The job was suspended because a suspend threshold limit was exceeded on the execution host while the job was in the STARTING state.

STARTING-SUSPENDED-PREEMPTION
: The job was suspended because a job from a higher priority queue preempted the job while it was in a STARTING state.

UNKNOWN
The job state is unknown.

Managing and Monitoring Queues

Managing Queues

Access the **Queues** area by selecting **Queues** from the left navigation area of the SAS Workload Orchestrator web interface.

The **Queues** area lists all the queues that have been defined for the grid. The table lists the state of the queue, the priority, the number of jobs pending and running on the queue, and the maximum number of jobs allowed on the queue and on each host in the queue. You can filter the list to locate queues that match specified criteria.

Queues are defined in the **Configuration** area of the SAS Workload Orchestrator web interface. See "Configuring Queues" on page 43 for information about defining queues.

To perform an action on a queue, select the entry for the queue in the table, and then select an action. Only grid administrators and queue administrators can perform actions. Here are the available actions:

(**Open**)
Opens a closed queue. The queue can accept new jobs.

(**Close**)
Closes an open queue. Any new jobs that are sent to a closed queue are rejected. Jobs that are already in the queue when it is closed might continue to be processed

Activate
Activates an inactive queue. An active queue attempts to schedule jobs that are in the queue to hosts for processing.

Inactivate
Makes a queue inactive. If the queue is also open, an inactive queue can still accept jobs, but none of the jobs in the queue can be processed.

Viewing Queue Details

To view detailed information about a queue, click the name of the queue in the **Queues** table.

The ***Queue_name*** window displays configuration details for the queue and a list of the jobs that are running on the queue.

The left side of the window displays the configuration parameters for the queue.

The right side of the window displays a list of jobs that are running or that have run on the queue. The table lists the name and ID of the job, the state of the job, the

user who submitted to job, the host on which the job is running, and the start and end time of the job.

You can change the state of the queue from this window. See “Managing Queues ” on page 68 for details about the queue states.

Click ⁝☰ or **Close** to return to the **Queues** table.

Filtering the Queue Table

You can filter the contents of the **Queues** table in order to locate queues that meet specified criteria.

Enter text in the **Filter** field at the top of the page to find queues that contain the specified string. The table displays any queues whose name matches the supplied string.

To display a selection of queues that match a specified set of criteria, use the **Advanced Filter** fields. You can specify multiple filter criteria. Here are the available filters:

State
: Displays queues whose state matches the selected combination of states (open or closed and active or inactive). The filter lists only those states that are present in the list of queues. For example, if there are no queues that have a state of OPEN-INACTIVE, that combination is not included in the filter list.

Priority
: Displays queues whose priority falls within the selected range. The upper and lower values for the range reflect the highest and lowest priority values that are defined for queues on the grid.

Jobs Pending
: Displays queues that have the number of jobs pending that fall within the selected range. Because the range reflects the highest and lowest number of pending jobs in queues, the values for the range will change over time.

Jobs Running
: Displays queues that have the number of jobs running that fall within the selected range. Because the range reflects the highest and lowest number of running jobs in queues, the values for the range will change over time.

Maximum Jobs
: Displays queues whose maximum number of jobs that can be running on the queue at a time fall within the selected range. A value of -1 specifies that the maximum number of jobs for the queue is not defined.

Maximum Jobs Per Host
: Displays queues whose maximum number of jobs that can be running on a host in the queue at a time fall within the selected range. A value of -1 specifies that the maximum number of jobs for the host in the queue is not defined.

After you specify the filter criteria, click **Apply**. The queues that match the selected filter criteria are displayed in the Queues table, and the filter fields are hidden. Click ▽ to display the filter fields.

Managing and Monitoring Hosts

Managing Hosts

Access the **Hosts** area by selecting **Hosts** from the left navigation area of the SAS Workload Orchestrator web interface.

The **Hosts** area lists all the hosts on the grid. The table in the **Hosts** area lists the host name, state of the host, the priority, the maximum number of jobs allowed, the number of jobs running and pending on the host, and the host's operating system. You can filter the list to locate hosts that match specified criteria. See "Host States" on page 72 for a list of the possible host states.

To perform an action on a host, select the entry for the host in the table, and then select an action. You can also select the check boxes next to multiple hosts in order to perform an action on all the selected hosts. Only administrators can perform actions. Here are the available actions:

(**Open**)
: Enables jobs to be processed on the host.

(**Close**)
: Prevents jobs from processing on the machine. Closing a host is useful when you want to remove the host from the grid for maintenance

Viewing Host Details

To view detailed information about a host, click the name of the host in the **Hosts** table.

The ***Host_ID*** window displays configuration details for the host and a list of the jobs that are running on the host.

The information about the host is organized into these tabs:

General
: Basic information about the host machine, such as operating system, number of cores, and memory.

CPU
: Detailed information about the host's CPU, including number of cores allocated, CPU rating, and CPU utilization.

Jobs (Overview)
: A summary of information about the jobs on the host, including the maximum number of jobs allowed, the number of jobs that are running, and the number of jobs that are suspended. The table also displays the CPU run queue length averaged over the course of 15 seconds, 1 minute, and 15 minutes.

Jobs (Running)
A table of currently running jobs, displaying the ID and the name, the state, the user who submitted the job, and the time at which the job was submitted, started, and ended.

Memory
Detailed information about the memory on the host, including the I/O rate, total memory, allocated memory, page rate, swap memory, amount of allocated swap memory, temporary memory, and allocated temporary memory.

OS
A summary of information about the host's operating system, including the IP address, state, status, and tags.

Resources
A list of values for all the static and dynamic resources for the host.

Schedule Thresholds
A list of any metrics that have been defined as schedule thresholds for the host. The table also lists the start value and end value for the metric. The start value is the value of the metric where the threshold goes into effect, which causes scheduling to stop on the host. The end value is the value of the metric where the threshold stops having an effect, which causes scheduling to resume on the host.

Suspend Thresholds
A list of any metrics that have been defined as suspend thresholds for the host. The table also lists the start value and end value for the metric. The start value is the value of the metric where the threshold goes into effect, which causes the lowest priority jobs to be suspended on the host. The end value is the value of the metric where the threshold stops having an effect, which causes suspended jobs to resume processing on the host.

Click ☰ or **Close** to return to the **Hosts** table.

Filtering Hosts

You can filter the contents of the **Hosts** table in order to locate hosts that meet specified criteria.

Enter text in the **Filter** field at the top of the page to find hosts that contain the specified string. The table displays any hosts whose name matches the supplied string.

To display a selection of hosts that match a specified set of criteria, use the **Advanced Filter** fields. You can specify multiple filter criteria. Here are the available filters:

State
Displays hosts whose state matches the selected combination of states. The filter lists only those states that are present in the list of hosts.

Operating System
Displays hosts that use the selected operating system.

Maximum Jobs Allowed
Displays hosts whose maximum number of jobs that can be running at a time fall within the selected range.

Jobs Running
Displays hosts whose number of currently running jobs fall within the selected range.

Jobs Suspended
Displays hosts whose number of suspended jobs fall within the selected range.

After you specify the filter criteria, click **Apply**. The hosts that match the selected filter criteria are displayed in the Hosts table, and the filter fields are hidden. Click ▽ to display the filter fields.

Host States

Host state is a combination of the *scheduling_state* and the *scheduling_status* values of the host. The *scheduling_state*, which is specified by the administrator, is either OPEN or CLOSED. This value indicates whether the administrator has enabled jobs to be scheduled to the host. The *scheduling_status* value is dynamically determined. It specifies whether jobs can actually be scheduled to the host.

See "Managing Hosts" on page 70 for information about changing the value of *scheduling_state*.

For example, if the host state is OPEN-FULL, the administrator has enabled jobs to be scheduled to the host (*scheduling_state* of OPEN), but the *scheduling_status* of FULL indicates that the host is already processing as many jobs as it can. A host state of CLOSED-OK indicates that the administrator has prevented jobs from being scheduled to the host (*scheduling_state* of CLOSED) although the host is available to process jobs (*scheduling_status* of OK).

Here are the possible host-state value combinations:

scheduling_state-UNKNOWN
A host enters the UNKNOWN state when it has not contacted the master for a period of time (specified by the **Host unknown** configuration property) after being in the INACTIVE state. The default value of **Host unknown** is 900 seconds (15 minutes).

For example, consider a case where **Host inactive** is set to 60 seconds and **Host unknown** is set to 120 seconds. If a host does not contact the master in 60 seconds, it goes into the INACTIVE state. If the host does not contact the master in an additional 120 seconds (for a total of 180 seconds), it goes into the UNKNOWN state. If a host goes into the UNKNOWN state and it is not a master candidate, it is removed from the host list. See "Configuring SAS Workload Orchestrator General Options" on page 41 for information about specifying this value.

scheduling_state-INACTIVE
The host has not contacted the master in the amount of time that is specified by the **Host inactive** configuration property. The default value of **Host inactive** is 60 seconds (one minute). See "Configuring SAS Workload Orchestrator General Options" on page 41 for information about specifying this value.

scheduling_state-OK
The host is active and is ready for jobs.

scheduling_state-OVERLOADED
A scheduling threshold or suspend threshold has been exceeded on the host.

scheduling_state-FULL
The maximum number of jobs are running on the host.

scheduling_state-NO_SLOTS
The host cannot run jobs because no more slots are available.

scheduling_state-UNLICENSED
The host has an invalid license.

Managing and Monitoring High Availability Services

Managing High Availability Services

Access the **Services** area by selecting **Services** from the left navigation area of the SAS Workload Orchestrator web interface.

The **Services** area lists all the defined high availability (HA) services. The table lists the name of the service, the state, whether the service is enabled or disabled, the user ID under which the service runs, and the active instances of the service. You can filter the list to locate services that match specified criteria. See "Service States" on page 74 for the possible service states.

HA services are defined in the **Configuration** area of the SAS Workload Orchestrator web interface. See "Defining High Availability Services on a SAS Grid Manager Grid" on page 89 for information about defining HA services.

To perform an action on an HA service, select the entry for the service in the table, and then select an action. Here are the available actions:

■ (**Disable**)
Disables an enabled service. All instances of the service are stopped.

▷ (**Enable**)
Enables a disabled service.

Filtering High Availability Services

You can filter the contents of the **Services** table in order to locate High Availability (HA) services that meet specified criteria.

Enter text in the **Filter** field at the top of the page to find services that contain the specified string. The table displays any hosts whose name matches the supplied string.

To display a selection of HA services that match a specified set of criteria, use the **Advanced Filter** fields. You can specify multiple filter criteria. Here are the available filters:

State
Displays services with a selected state.

Active instances
Displays services that have a number of instances that fall within the selected range

Maximum Wait
Displays services that have a maximum wait time that falls within the specified range.

User
Displays services that run under the credentials of the specified user.

After you specify the filter criteria, click **Apply**. The services that match the selected filter criteria are displayed in the Services table, and the filter fields are hidden. Click ▽ to display the filter fields.

Service States

Here are the possible service states:

ADDED
The service has been added, but no instances have been launched.

STOPPED
The service has been stopped

RUNNING
The specified number of instances are running.

RUNNING-PARTIAL
The service is running, but with fewer than the specified number of instances.

LAUNCH-ERROR
The service failed to run due to a launch error.

RUN-ERROR
The service failed to run due to an error with the script for the service.

Viewing Log Messages

Managing Messages

Access the **Logs** area by selecting **Logs** from the left navigation area of the SAS Workload Orchestrator web interface.

The **Logs** area includes a table of log messages that have been produced by the grid-related loggers. The table separates the logger, message level, date, and message in order to make it easier to view. To view the messages in the form in which they are written to the log, click ▤. To return to the table view, click ▤. You can filter the list to locate services that match specified criteria.

Click ⊟ (**Download Log**) to download the displayed log messages to a text file.

Filtering Log Messages

You can filter the contents of the **Logs** table or listing in order to locate log messages that meet specified criteria.

Enter text in the **Filter** field at the top of the page to find messages that contain the specified string.

To display a selection of services that match a specified set of criteria, use the **Advanced Filter** fields. You can specify multiple filter criteria. Here are the available filters:

Logger
Displays services that are produced by a specified logger.

Level
Displays only messages that are at the selected level.

Threshold
If you select a **Level** filter, the messages that are displayed are at the selected level as well as at higher levels. For example, if you select WARN in the **Level** filter and also select the **Threshold** filter, the table displays messages that have the levels of WARN, ERROR, and FATAL.

Date Range
Displays messages that were written to the log between the specified start date and end date.

Host
Displays only log messages that are produced by the specified host.

After you specify the filter criteria, click **Apply**. The log messages that match the selected filter criteria are displayed in the Logs table, and the filter fields are hidden. Click ▽ to display the filter fields.

Using the SAS Workload Orchestrator Utilities

Specifying Logging Levels

You can change the logging level for grid-related loggers. Follow these steps:

1. In the SAS Workload Orchestrator web interface, select **Utilities** in the left navigation area. The Utilities page is displayed, and the **Set Log Levels** tab is selected by default. The **Set Log Levels** tab lists all the defined loggers that are used by the grid hosts.
2. Select the check box for the logger whose logging level you want to change. You can select multiple loggers.
3. Select the **Set Log Level** menu.

4 Select the logging level to use for the selected logger. Selecting a value of NULL specifies that the logger inherits the logging level of its parent logger.

5 Confirm the change when prompted.

Updating the License File

When you update your SAS license, you must also follow a separate process to update the SAS Grid Manager license. Follow these steps to update the license file:

1 In the SAS Workload Orchestrator web interface, select **Utilities** in the left navigation area.

2 In the Utilities page, select the**Update License** tab.

3 Click **Choose a file upload** and navigate to the location of your SAS Grid Manager license file. Select the file in the form `SAS94_xxx.txt`, where *xxx* specifies the grid platform.

4 Click **Update License**.

Using SAS Environment Manager

Overview

If your grid uses SAS Workload Orchestrator, SAS Environment Manager provides an agent plug-in that enables you to monitor and manage a SAS grid. The plug-in uses continuously collected metric data to monitor the performance of the grid, grid servers, and grid queues in order to graph changing metric data and to generate alerts.

The SAS Workload Orchestrator agent plug-in for SAS Environment Manager provides metric data in a SAS Workload Orchestrator environment for the grid, individual grid hosts, and grid queues. It uses that data to perform these functions:

- display the current state of grid resources
- graph the data over time, providing a historical view and enabling you to see how the data changes
- create alerts that notify you whenever a specified measurement reaches a specified state

See Installing and Configuring SAS Environment Manager in a SAS Grid Environment with a Shared Configuration Directory for information about installing and configuring SAS Environment Manager so that it can monitor grid resources.

Configuring the SAS Workload Orchestrator Agent Plug-in

Before the SAS Workload Orchestrator agent plug-in can discover the grid resources, it must be configured. Follow these steps to configure the plug-in:

1. Log on to SAS Environment Manager with an administrative user ID.
2. Select **Resources** ⇨ **Browse** ⇨ **Servers**. The **SAS Workload Orchestrator** entry in the table contains a gray question mark icon in the **Availability** column, which indicates that it has not been configured. Select the **SAS Workload Orchestrator** entry in the table to display the Resource Details page.
3. Select **Tools Menu** ⇨ **Configure Server** to display the Configuration Properties page.
4. Specify the grid administrator's user ID in the **swo.admin.user** field and password in the **swo.admin.pwd** field.
5. Ensure that the **Auto-Discover Hosts and Queues** check box is selected.
6. Click **OK**.

The SAS Workload Orchestrator agent plug-in retrieves the master host information from metadata. If you use the SAS Workload Orchestrator web interface to change the master host or master host candidates, you must also manually change the master host information in metadata in order for the agent plug-in to operate. See "Updating Master Hosts in Metadata" on page 55 for information about this process.

Monitoring Grid Resources

Discovering Grid Resources

After the agent plug-in is configured, SAS Environment Manager will automatically discover the service resources that are associated with the SAS Workload Orchestrator grid. By default, the process to discover new service resources (such as grid hosts and queues) runs every 24 hours. Regardless of whether new or changed service resources are being discovered, up to 24 hours might be necessary for the discovery process to be completed.

To circumvent the 24-hour time period, you can restart the SAS Environment Manager agent, which causes the discovery process to start immediately. If you want to change the interval at which new or changed resources are discovered:

1. Modify the `SASConfig/Lev1/Web/SASEnvironmentManager/agent-5.8.0-EE/conf/agent.properties` file.

2 In this file, modify the parameter `autoinventory.runtimeScan.interval.millis=`*`interval`*, where *`interval`* is the autodiscovery interval in milliseconds. The default value is `86400000`.

3 Restart the SAS Environment Manager agent after you change the value of the parameter.

After the resources are discovered, SAS Environment Manager starts collecting metric data that you can use to monitor the health and operation of your SAS grid. You can view data for the entire grid and for individual grid hosts and queues.

Viewing Availability and Metrics for a Grid

To view availability and metrics for the grid, follow these steps:

1 In SAS Environment Manager, select **Resources** ⇨ **Browse** and select **Servers**. A table of server resources appears.

2 Locate and select the **SAS Workload Orchestrator** entry in the table.

The details page displays availability and metric data for the grid. See "Using the Details Page for a Grid Resource" on page 80 for more information.

Here are the metrics that are collected for the grid:

- Availability
- CPU Utilization
- Disk IO Rate
- IO Rate
- Jobs
- Jobs Finished
- Jobs Pending
- Jobs Running
- Jobs Suspended
- Max Jobs Allowed
- Max Physical Memory
- Max Swap Space
- Max Temp Space
- Net IO Rate
- Number of Cores
- Run Queue Average 15m (threads per 15 minutes)
- Run Queue Average 15s (threads per 15 seconds)
- Run Queue Average 1m (threads per minute)
- Used Physical Memory
- Used Swap Space
- Used Temp Space

Viewing Availability and Metrics for a Grid Host

To view availability and metrics for individual hosts on a SAS Workload Orchestrator grid, follow these steps:

1. Select **Resources** ⇨ **Browse**, select **Services**, and search for a host name. You can also select **SAS Workload Orchestrator Host** in the **All Service Types** field, and then select one of the hosts in the table.
2. Locate and select a host entry in the table. The table contains a separate entry for each host on the grid.

   The details page displays availability and metric data for the selected host. See "Using the Details Page for a Grid Resource" on page 80 for more information.

Here are the metrics that are collected for each grid host:

- Availability
- CPU Utilization
- Disk IO Rate
- IO Rate
- Is Master
- Jobs Finished
- Jobs Running
- Jobs Suspended
- Max Jobs Allowed
- Net IO Rate
- Run Queue Average 15m (threads per 15 minutes)
- Run Queue Average 15s (threads per 15 seconds)
- Run Queue Average 1m (threads per minute)
- Used Physical Memory
- Used Swap Space
- Used Temp Space

Viewing Availability and Metrics for a Grid Queue

To view availability and metrics for a SAS Workload Orchestrator queue, follow these steps:

1. In SAS Environment Manager, select **Resources** ⇨ **Browse**, **Services**, and search for a queue name. You can also select **SAS Workload Orchestrator Queue** in the **All Service Types** field, and then select one of the queues in the table.
2. Locate and select a queue entry in the table. The table contains a separate entry for each queue that you have defined.

   The details page displays availability and metric data for the selected queue. See "Using the Details Page for a Grid Resource" on page 80 for more information.

Here are the metrics that are collected for each grid queue:

- Availability
- Jobs Finished
- Jobs Pending
- Jobs Running
- Jobs Suspended
- Pending Time

Using the Details Page for a Grid Resource

The details page for a grid resource displays an availability indicator and graphs for selected metrics for the selected resource. You can also view a table of all collected metrics for the resource.

Reading the Availability Bar

The **Monitor** tab on the details page for a grid resource contains an **Availability** bar, which enables you to check the availability of the grid or a grid host at a glance. The bar displays a color-coded indicator that represents the resource's availability during a time slice (the length of which depends on the display range that you select). The percentage of time that the resource was available is displayed at the right side of the availability bar. Here are the meanings of the color codes:

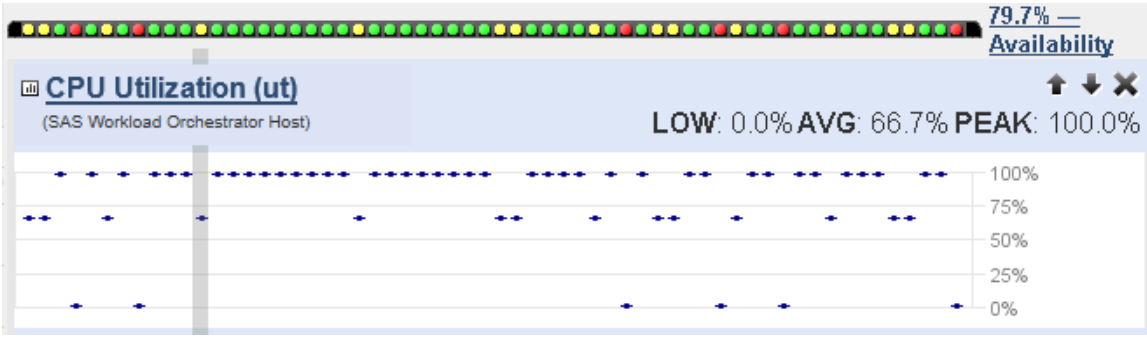
Green
: 100% availability

Yellow
: Partial availability (between 0% and 100%)

Red
: 0% availability

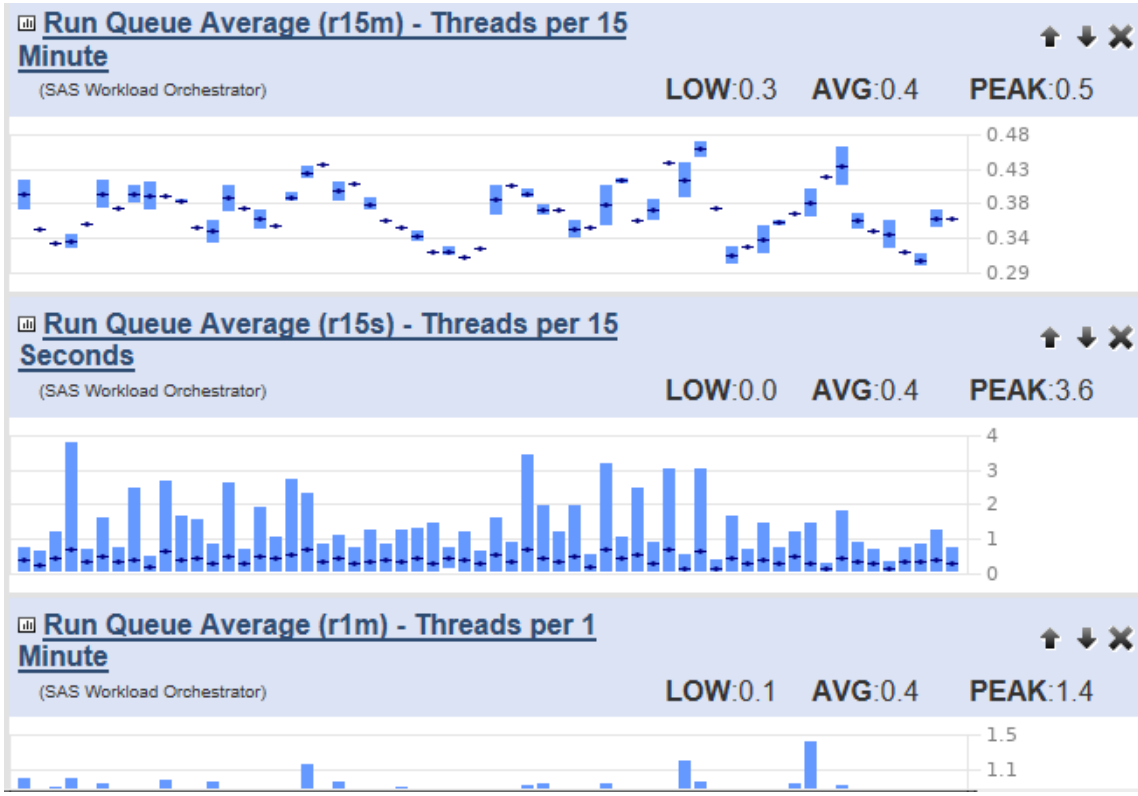
Click the percentage availability at the end of the **Availability** bar to view a graph of the availability information.

Clicking a dot in the **Availability** bar highlights the time slice in the charts beneath the bar, which helps you diagnose availability problems.



Viewing Metric Graphs

The **Monitor** tab on the details page for a grid resource displays a set of charts that contain data for each collected metric over a specified time period. To change the time period that is displayed in the graphs, select a value in the **Metric Display Range** field. The default time period is eight hours.



Viewing Metric Data

To view detailed metric data, click **Metric Data**. The details page changes to display the metric data in table form. By default, the table displays only the available data. To see all the collected metric data, click the arrow next to **Show All Metrics**. By

default, the metric data refreshes every two minutes. To change the refresh interval, click a value next to **Metrics Refresh**.

Indicators | Metric Data

Show All Metrics: | Last Updated: 11/01/2018 04:40:08 PM | Metrics Refresh: 1 min | 2 min | 5 min | OFF

Availability	Alerts	OOB	LOW	AVG	PEAK	LAST	Collection Interval
Availability	0	0	-	100.0%	-		00:01:00
Utilization							
CPU Utilization (ut)	0	0	0.5%	2.4%	16.7%	2.9%	00:01:00
Disk IO Rate - MB per sec	0	0	0.0	0.1	0.5	0.1	00:01:00
IO Rate - MB per sec	0	0	0.1	0.2	0.9	0.2	00:01:00
Jobs	0	0	0.0	0.0	0.0	0.0	00:01:00
Jobs Finished	0	0	0.0	0.0	0.0	0.0	00:05:00
Jobs Pending	0	0	0.0	0.0	0.0	0.0	00:01:00
Jobs Running	0	0	0.0	0.0	0.0	0.0	00:01:00
Jobs Suspended	0	0	0.0	0.0	0.0	0.0	00:05:00
Max Jobs Allowed	0	0	16.0	16.0	16.0	16.0	01:00:00
Max Physical Memory	0	0	-	-	-	15.6 GB	01:00:00
Max Swap Space	0	0	-	-	-	7.8 GB	01:00:00
Max Temp Space	0	0	-	-	-	216.2 GB	01:00:00
Net IO Rate - MB per sec	0	0	0.0	0.1	0.3	0.1	00:01:00
Run Queue Average (r15m) - Threads per 15 Minute	0	0	0.3	0.4	0.5	0.4	00:05:00
Run Queue Average (r15s) - Threads per 15 Seconds	0	0	0.0	0.4	3.6	0.5	00:01:00
Run Queue Average (r1m) - Threads per 1 Minute	0	0	0.1	0.4	1.4	0.5	00:01:00
Used Physical Memory	0	0	14.4 GB	14.4 GB	14.5 GB	14.5 GB	00:05:00
Used Swap Space	0	0	338.3 MB	360.2 MB	366.2 MB	338.3 MB	00:05:00
Used Temp Space	0	0	112.6 GB	112.7 GB	112.9 GB	112.9 GB	00:05:00

Creating Alerts

SAS Environment Manager enables you to create an alert for any of the metrics that are collected by the SAS Workload Orchestrator agent plug-in. An alert is a user-defined type of event that indicates a critical condition in a selected resource. When an alert occurs, it must be acknowledged. Also, alerts remain in the list until they are marked as having been fixed. You can define escalation schemes to identify the actions that happen if an alert is not fixed within a specified time.

For example, you can create an alert that occurs whenever availability of the grid falls below a certain percentage or when the number of available job slots falls below a certain threshold. See Working with Resource Alerts in *SAS Environment Manager2.5: User's Guide* for more information about alerts.

To create an alert, follow these steps:

1. Select **Resources** ⇨ **Browse** and either **Servers** (for the grid) or **Services** (for the grid nodes). Click the alert icon (which is located within the group of three icons to the left of the entry in the table) for the resource. The Alerts page for the resource appears.

2. Click **New** to display the New Alert page.

3. In the New Alert page, specify the metric for which you want to create an alert and the condition for the metric that causes the alert. The page also enables you to define other criteria for the alert, such as how often the alert can occur. See Defining an Alert in *SAS Environment Manager2.5: User's Guide* for more information about defining an alert.

Using the SAS Workload Orchestrator Administration Utility

You can perform many of the tasks of managing and monitoring a SAS Workload Orchestrator grid with the SAS Workload Orchestrator Administration Utility, which is an administration command-line interface.

The SAS Workload Orchestrator Administration Utility is located in the directory ***SASConfig*/Lev1/Applications/GridAdminUtility**. This directory contains the script for running the utility (`sas-grid-cli.sh`), the configuration information for the utility, and the logging configuration files for the utility (logconfig.xml and logconfig-trace.xml).

Here is the syntax of the command line to run the SAS Workload Orchestrator Administration Utility:

```
sas-grid-cli (connection options) (authentication options)
[global options <arguments>] command [command options] [arguments]
```

Run the `sas-grid-cli -h` command to view a list of possible options, commands, and arguments.

Use these commands to manage and monitor a SAS Workload Orchestrator grid:

Connection options

`--mhost` *`hostname`*
: specifies the master host

`--mport` *`port`*
: specifies the port for the master host

`--mScheme <http | https>`
: specifies the http protocol to use when communicating with the master host. The default value is `http`.

`--mURL` *`host_URL`*
: specifies the URL of the master host. The URL must include the scheme (http or https), the host name, and the host port. Specify either `--mURL` or a combination of `--mScheme`, `--mhost`, and `--mport`.

Authentication options

If both `--grid-user` and `--grid-password` are specified, the supplied user name and password are used to authenticate to the grid. If neither of these options is specified, and `--sspi` is specified, Kerberos authentication is used. If neither of these options is specified, authinfo credentials are used for authentication, regardless of whether the `--authinfo` option is specified. If the location of the authinfo file is not specified using the `--authinfo` option, the default authinfo location is used. If you specify both `--authinfo` and the combination of `--grid-user` and `--grid-password`, `--authinfo` is used and a warning message is returned.

`--authinfo` *`authinfo_pathname`*
: specifies that credentials from an authinfo file are used to connect to the grid. If a location is not specified, the default location is used.

`--grid-user` *`username`*
specifies a user name to connect to the grid

`--grid-password` *`password`*
specifies the password for the grid user

`--sspi`
specifies that Kerberos authentication is used to authenticate connections to the grid.

Global options

`--help`
displays the command help

`--debug | --trace`
enables debug or trace-level logging

`--log` *`filename`*
enables logging to the specified file

`--logconfigloc` *`filepath`*
specifies the location of a logging configuration file

Commands

`gridinfo`
returns a list of general information about the grid

`show-master`
returns the name of the current master host

`show-hosts --host <host1>, <hostn> <--output json>`
returns information about the specified hosts or all hosts. By default, information is returned in tabular format. Specify `--output json` to return the information in JSON format.

```
show-jobs --id <job_id_1>, <job_id_n> <--output json>
show-jobs --user <user>
show-jobs --state PENDING | RUNNING | SUSPENDED | FINISHED |
ARCHIVED | ALL <--output json>
show-jobs --queue <queue> <--output json>
show-jobs --host <host> <--output json>
show-jobs --firstJobID <job_id> <--output json>
show-jobs --lastJobID <job_id> <--output json>
show-jobs --firstEndTime <date_time> <--output json>
show-jobs --lastEndTime <date_time> <--output json>
show-jobs --limit <max_number_of_jobs> <--output json>
```
returns a list of all jobs or specified jobs that meet the selection criteria. By default, information is returned in tabular format. Specify `--output json` to return the information in JSON format.

`show-services --id <service_1>, <service_n> <--output json>`
returns information about the specified high availability services or all high availability services. By default, information is returned in tabular format. Specify `--output json` to return the information in JSON format.

`show-queues --queue <queue_1>, <queue_n> <--output json>`
returns information about the specified queue or all queues. By default, information is returned in tabular format. Specify `--output json` to return the information in JSON format.

`open-host --host <host>, <host>`
opens a specified host

`close-host --host <host>, <host>`
closes a specified host

`suspend-job --id <job-id>, <job-id>`
suspends a specified job

`resume-job --id <job-id>, <job-id>`
resumes processing for a specified suspended job

`cancel-job --id <job-id>, <job-id>`
cancels a specified job

`enable-service --service <service>, <service>`
enables a specified high availability service

`disable-service --service <service>, <service>`
disables a specified high availability service

`open-activate-queue --queue <queue>, <queue>`
opens and activates a specified queue. An open and active queue can accept and process jobs.

`open-inactivate-queue --queue <queue>, <queue>`
inactivates an open queue. An inactive open queue can still accept jobs, but none of the jobs in the queue can be processed.

`close-activate queue --queue <queue>, <queue>`
closes a specified queue but leaves the queue in an active state. A closed queue cannot accept any jobs that are sent to it, but jobs that are in the queue continue to process.

`close-inactivate queue --queue <queue>, <queue>`
closes and inactivates a specified queue. A closed and inactive queue cannot process or accept jobs.

`show-logger-levels [--host <host>, <host>]`
displays the logger levels for a specified host or all hosts

`set-logger-level --logger <loggername>, <loggername> --level <none|fatal|error|warn|info|debug|trace|all|null>`
sets a specified level for one or more specified loggers

`set-gridconfig --file <config_file_path>`
specifies that SAS Workload Orchestrator use a specified grid configuration file

`update-license --file <license_file_path`
updates the grid license using the specified license file

`update-passphrase --passphrase new_passphrase`
updates the grid passphrase using the specified new passphrase

```
show-log-lines --host <host>
show-log-lines --logger <logger>
show-log-lines --level trace|debug|info|warn|error|fatal
show-log-lines --threshold trace|debug|info|warn|error|fatal
show-log-lines --starttime <date_time>
show-log-lines --endtime <date_time>
```
returns all lines of the log or lines that match the specified criteria

Using Control Scripts

The `<config>/Levn/Grid` directory contains scripts, which enable you to control the SAS Workload Orchestrator daemons. Here are the control scripts:

gridStart
: Starts all daemons on all grid nodes. You must be an administrator to run this command.

gridStop
: Stops all daemons on all grid nodes. You must have access to the SAS Workload Orchestrator shared directory, and you must have Read access to the SAS Workload Orchestrator master state file. On Windows, you must be an administrator, and you must be able to start services on other machines. On UNIX, you must be an administrator, and you must be able to use SSH without passwords to connect to other machines. The Requests Python library is required in order to use this command.

4

High Availability in SAS Grid Manager

High Availability on a SAS Grid Manager Grid

Your organization might have services and long-running SAS programs that are critical to your operations. The services must be available at all times, even if the servers that are running them become unavailable. SAS programs must complete in a timely manner, even if something happens to cause them to fail. For a SAS program that takes a long time to run, this means that the program cannot be required to restart from the beginning if it has ended prematurely.

SAS Grid Manager provides high availability (HA) through these capabilities:

- Multi-machine architecture. Because of how a SAS grid is configured and operates, there is no single point of failure. Because jobs are processed on the available grid nodes, if a node becomes unavailable, other nodes can take over the workload.
- Master host failover. During configuration, you can define multiple hosts to be master host candidates. During operation, the first host in the list of candidates operates in the role of master. If a call to the master fails because of a connection problem, the assumption is that there is a problem with the master host, so the other master candidates ping each other. The first candidate that responds takes over the role of master.

- Critical service failover. There are certain services and processes that are critical to the operation of SAS applications on the grid and that must always be available (for example, the SAS Metadata Server). You can define the hosts on which the service runs and the number of instances of the service that are always available. If the service fails, SAS Workload Orchestrator can automatically restart the service on the next specified host, ensuring that the service remains available.
- Automatic SAS program failover. If a long-running SAS job fails before completion, rerunning it from the beginning can cause a loss of productivity. You can use the SAS Grid Manager Client Utility to specify that the job is restartable. This means that a failed job restarts from the last successful SAS procedure, DATA step, or labeled section. This capability uses the SAS checkpoint and restart functions to enable failed jobs to complete without causing delays. You can also use attributes in the queue definitions on the grid to automatically restart any job that terminates due to host failure. Using these options together ensures that critical SAS programs always run successfully and in a timely manner, even if they encounter problems.

Master Host Failover

The master host controls distribution of jobs to hosts on the grid. Although only one host at a time can be the master, you can specify multiple hosts as masters to provide failover capability. The list of master host candidates is stored in a configuration file. The master periodically sends this configuration file to the other hosts. The state of all hosts on the grid is stored in a shared directory.

During grid operation, all hosts periodically send their status to the master. If a status update fails because the host is unreachable, the master is assumed to have failed. If this happens, the other master candidates initiate a search for a new master host by trying to reach the other master candidates. The master candidates poll each other in the order in which they are listed in the configuration. The first master candidate that responds takes over as the new master host and restores the state of the grid from the information in the master state file and the job history file.

For information about defining master hosts, see “Defining a New Host Type” on page 50.

High Availability Services

Periodically, the host on which a high availability service is running checks the status of the service. If the service has a permanent error or has accumulated too many temporary errors, the host restarts the service. If the service cannot be restarted or has been restarted too many times, the host stops the service and sends that information to the master host.

Every 15 seconds, the master host checks the status of the service. If there are fewer running instances of the service than are required, the master host finds an

active host on which the service is defined but is not running and sends a request to start the service on that host. The master host continues this process until the required number of running instances of the service is met or there are no more active and available hosts.

Defining High Availability Services on a SAS Grid Manager Grid

Follow these steps to set up a high availability service:

1. In the SAS Workload Orchestrator web interface, select **Configuration** in the left navigation area.
2. On the Configuration page, select the **Services** tab.
3. Select **New service** and specify a name for the service.
4. In the **Number of instances** field, specify the number of instances of the service that must be available. Specify a value of 1 if the service should be restarted on another host if the service fails (for example, the SAS Metadata Server). Specify a value that is greater than 1 if multiple simultaneous instances of the service (such as a load-balanced object spawner) need to run.
5. Select **Auto start** to specify that the service should automatically start when the grid starts. If **Auto start** is not specified, you must enable the service through the SAS Workload Orchestrator web interface.
6. Specify the hosts on which the service can run. You can specify values for **Host names** or **Host types**. You do not have to specify both fields.

   If a new instance of the service needs to be started, the master host attempts to start the service on these hosts, host groups, or host types in the order in which the master stores the host information.
7. In the **Script file path** field, specify the complete path and file name for the script that is used to control the service. The script runs with one parameter, which can be `start`, `stop`, `status`, or `restart`. The host periodically runs this script with the `status` parameter to determine the status of the service. The host also runs the script with the `start`, `restart` or `stop` parameters to start, restart, or stop the service.

   See “Sample Service Scripts” on page 321 for sample scripts.
8. Specify the user ID and password under which the service runs. The password can be a value that is encoded by SAS.
9. Select to save the configuration.

5

Troubleshooting SAS Grid Manager

Fixing Issues with SAS Workload Orchestrator

If you are using SAS Grid Manager in a Windows environment, you might see this message in the SAS Workload Orchestrator log:

```
ERROR [00000012] App.Grid.SGMG.Log.Status.Sigar (TKWINutil.c:46) -
Unable to get information about processes on the system.
```

The log is located in the directory ***SASconfig*\Levn\Grid\Logs**. The log file is named using the convention `Grid_date_hostname_pid.log`. An example of a log file name is `Grid_2018-11-13_FSSVM37_5008.log`.

This message might appear several times in the log. The message indicates that SAS Workload Orchestrator is unable to obtain a list of current processes on the system. This can happen if the Process performance object is missing.

Follow these steps to resolve the problem:

> **IMPORTANT** Contact your system administrator before making changes to the registry.

1 Launch regedit.

2 Locate the registry key: **HKEY_LOCAL_MACHINE** ⇨ **SYSTEM** ⇨ **CurrentControlSet** ⇨ **Services** ⇨ **PerfProc** ⇨ **Performance**.

3 Locate the value Disable Performance Counters and ensure that it is set to 0 (zero). If it is set to anything other than 0, change it to 0.

4 Restart the Windows Management Instrumentation service.

5 Use the `gridStart` command to relaunch SAS Workload Orchestrator.

Enabling I/O Metrics in Linux

By default, SAS Workload Orchestrator running as the SAS installer on Linux does not have permission to read local disk I/O metrics for the processes that it manages. If the I/O metrics are important to your deployment, perform these steps:

1 Enable SAS Workload Orchestrator to find its dynamic load libraries. On each grid host, create the file **`/etc/ld.so.conf.d/sgmg.conf`**. In this file, add these lines:

```
SASHome/Secure/sasexe
SASRoot/sasexe
```

2 On each grid host, run this command as root:

```
ldconfig
```

3 If SAS Workload Orchestrator is installed and run locally, run this command as root:

```
setcap CAP_SYS_PTRACE,CAP_DAC_READ_SEARCH+ep SASRoot/utilities/bin/sgmg
```

4 If SAS Workload Orchestrator is installed and run over a network from a shared location (for example, using NFS or NAS), run these commands as root:

```
chown root:sas SASRoot/utilities/bin/sgmg
chmod 4755 SASRoot/utilities/bin/sgmg
```

To verify that the IO metrics are being collected:

1 In SAS Workload Orchestrator, select **Configuration** in the left navigation area.

2 On the Configuration page, select the **Queues** tab

3 Expand the entry for the **default** queue.

4 In the **Limits** area, click **Select an item** and select **MaxIoTotal** from the menu.

5 Click + and enter an arbitrarily large number (such as 1000000000).

6 Click ⊟ to save the configuration.

7 Submit a new job to the grid that will read or write to the host's local disk.

8 In SAS Workload Orchestrator, select **Jobs** in the left navigation area.

9 Select the job that you just submitted and view the job details. Verify that the value of maxIoTotal as greater than zero.

10 Edit the **default** queue definition, remove the entry for **MaxIoTotal** in the **Limits** area, and save the configuration.

PART 3

Grid Computing for SAS Using SAS Grid Manager for Platform

6

Planning and Configuring a SAS Grid Manager for Platform Grid Environment

SAS Grid Manager for Platform Installation and Configuration Overview

The process of configuring a SAS Grid Manager for Platform grid consists of two main tasks:

1. Installing and configuring the grid provider. Instructions for installing and configuring Platform Suite for SAS are found on the SAS website at Installing and Configuring a SAS Grid Environment.
2. Installing and configuring SAS products and metadata definitions on the grid. You can either install all SAS products on all machines on the grid or install different sets of SAS applications on sets of machines on the grid. However, Base SAS, SAS/CONNECT, and SAS Grid Manager for Platform must be installed on all grid machines. Using a grid plan file with the SAS Deployment Wizard guides you through the process of installing and configuring the SAS applications and metadata definitions on each machine on the grid. You must specify the same directory structure on all machines on the grid.

For information about performing a planned installation, see *SAS Intelligence Platform: Installation and Configuration Guide*.

Configuring the File Server

The central file server is a critical component of a grid environment. It is essential for each application on a grid node to be able to efficiently access data. Slowdowns caused by the performance of the file storage system could reduce the effectiveness and benefit of using a grid. The amount of storage required and the type of I/O transactions help determine the type of file storage system that best meets your needs. A shared file system is required if you want to use the restart functions of SAS Grid Manager Client Utility.

Assuming that the SAS jobs running on the grid perform an equal number of reads and writes, it is recommended that the file system be able to sustain 75–100 MB per second per core. This level can be adjusted up or down, depending on the level of I/O activity of your SAS jobs. For information about choosing and configuring a file system, see Shared File Systems: Determining the Best Choice for your Distributed SAS® Foundation Applications and Troubleshooting system performance problems: Shared/clustered file systems.

Installing Platform Suite for SAS

SAS Grid Manager for Platform includes Platform Suite for SAS from Platform Computing. The SAS website provides step-by-step instructions on installing and configuring the Platform Suite for SAS. These instructions are available from Installing and Configuring a SAS Grid Environment.

Information for installing Platform Suite for SAS is available for both Windows and UNIX platforms.

The installation process for Platform Suite for SAS installs these components:

- Platform Process Manager
- Platform LSF

Configuring the SAS Grid Manager for Platform Grid Control Server

After you install and configure Platform Suite for SAS, you can use the SAS Deployment Wizard to configure the grid control server. The SAS Deployment Wizard installs and configures these components:

Table 6.1 *SAS Deployment Wizard Grid Control Server Components*

Installed SAS Software Components	Configured SAS Software Components
■ SAS Foundation (including Base SAS and SAS/CONNECT) ■ SAS Management Console ■ Grid Manager Plug-in for SAS Management Console (see Note) ■ SAS Environment Manager	■ Platform Process Manager Server ■ Platform Web Services ■ SAS Environment Manager Server ■ SAS Grid Manager Agent Plug-in for SAS Environment Manager ■ SAS Grid Manager module for SAS Environment Manager ■ Grid Monitoring Server ■ SAS Application Server (SAS Logical DATA Step Batch Server, SAS Logical Grid Server, SAS Logical Workspace Server) ■ Object Spawner ■ Grid script file

Note: Starting in SAS 9.4M7, Grid Management Services (GMS) is removed from Platform Suite for SAS. GMS version 8.01, which was delivered with SAS 9.4M6, is not compatible with Platform Suite for SAS 11.1 (including Process Manager 10.2 and LSF 10.16), which was also delivered with SAS 9.4M6. Therefore, GMS and the Grid Manager plug-in for SAS Management Console are in Limited Support. See SAS Technical Support Services and Policies for more information about Limited Support.

If you are installing Platform Suite for SAS on a UNIX machine, you must source the profile.lsf file before you start the SAS Deployment Wizard. The `hostsetup` command that is used in the installation process for Platform LSF, starting with version 7, includes the ability to source the LSF profile from the default profile for all users. If this option was not used in the installation process or did not work correctly, you must use the following process instead. This process enables the SAS Deployment Wizard to locate the LSF addresource utility.

To source the profile.lsf file:

1 Start the LSF daemons. The easiest way to start the daemons is to reboot the computer on which Platform Suite for SAS is installed.

2 Using the default profile for the machine, source the profile file:

```
. LSF_Platform_directory/conf/profile.lsf
```

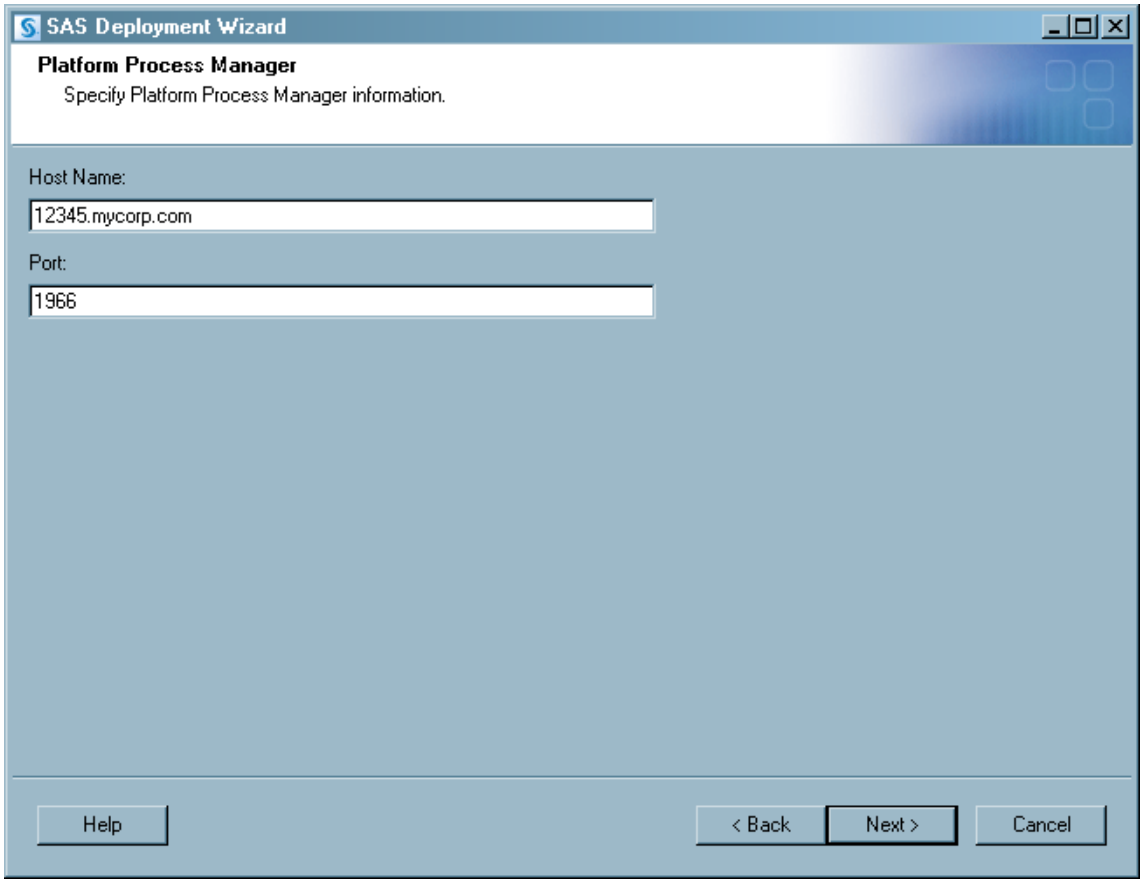
Replace *LSF_Platform_directory* with the name of the directory in which Platform LSF is installed. Note that the command starts with a period followed by a space.

The amount of user input that is required during the installation and configuration process depends on whether you choose an Express, Typical, or Custom install. For information about running the SAS Deployment Wizard, see *SAS Deployment Wizard User's Guide*.

An Express installation does not request any grid-specific information. Default values are used in all cases, so you must verify that these values match the values needed for your environment.

During an Express install, the Platform Process Manager information page enables you to specify the host name and port of the machine on which Platform Process Manager installed.

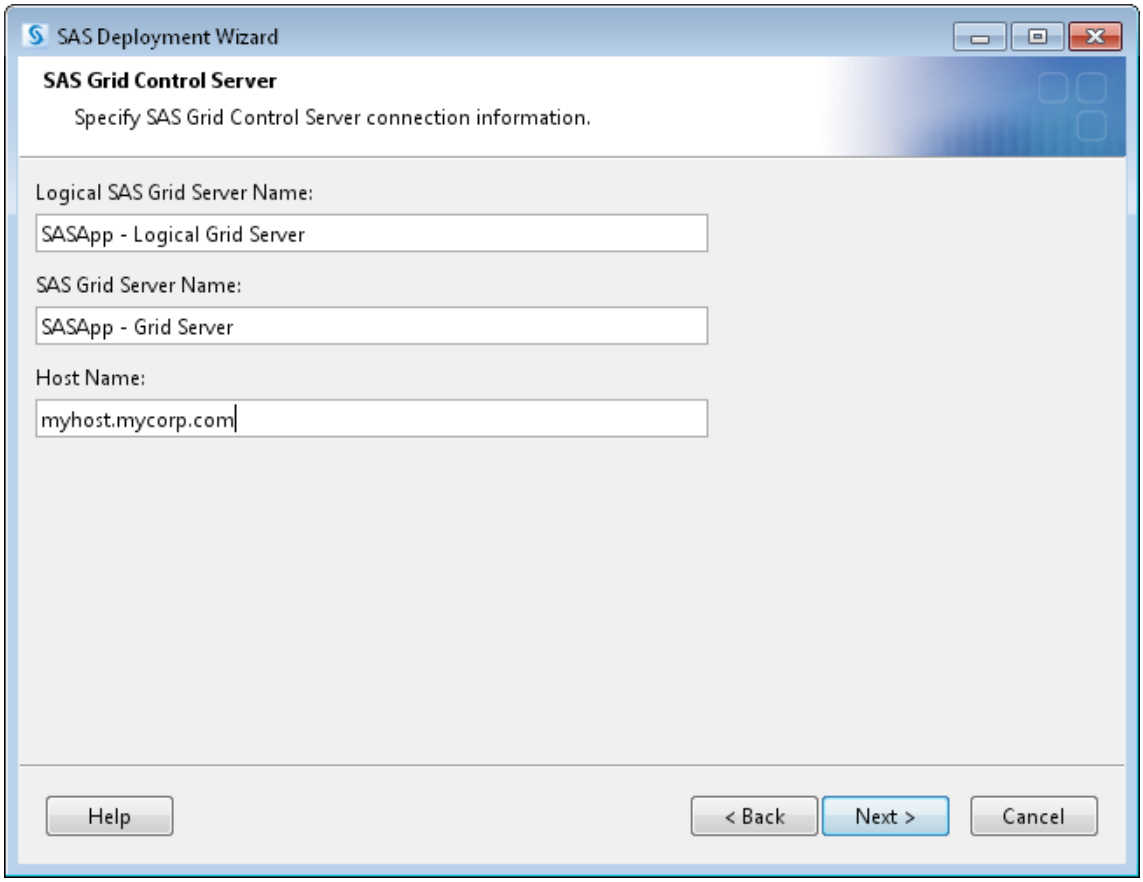
Figure 6.1 *Platform Process Manager Page for Express Install*



During the installation and configuration process for a Custom install, the SAS Deployment Wizard displays these pages that request grid-specific information:

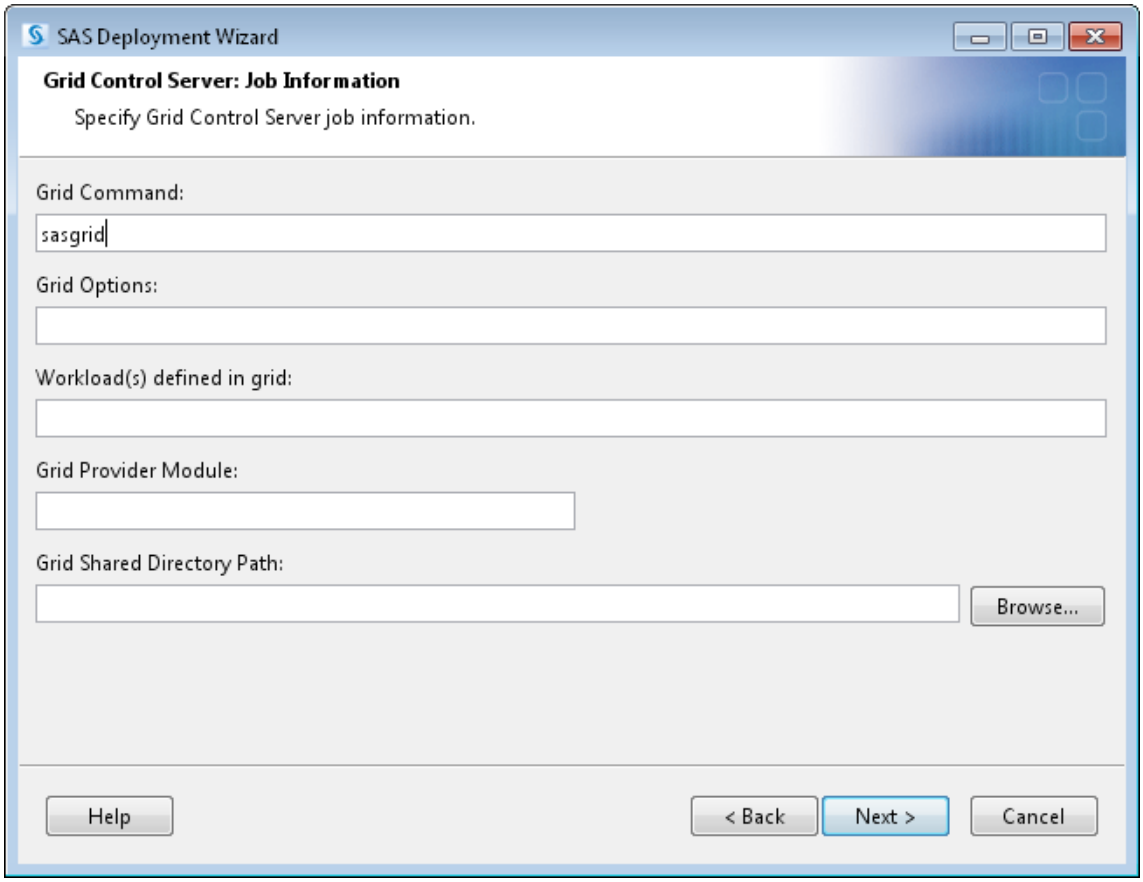
1 The SAS Grid Control Server page enables you to specify the name of the Logical SAS Grid Server and the SAS Grid Server. Specify the grid control server machine.

Figure 6.2 SAS Grid Control Server Page



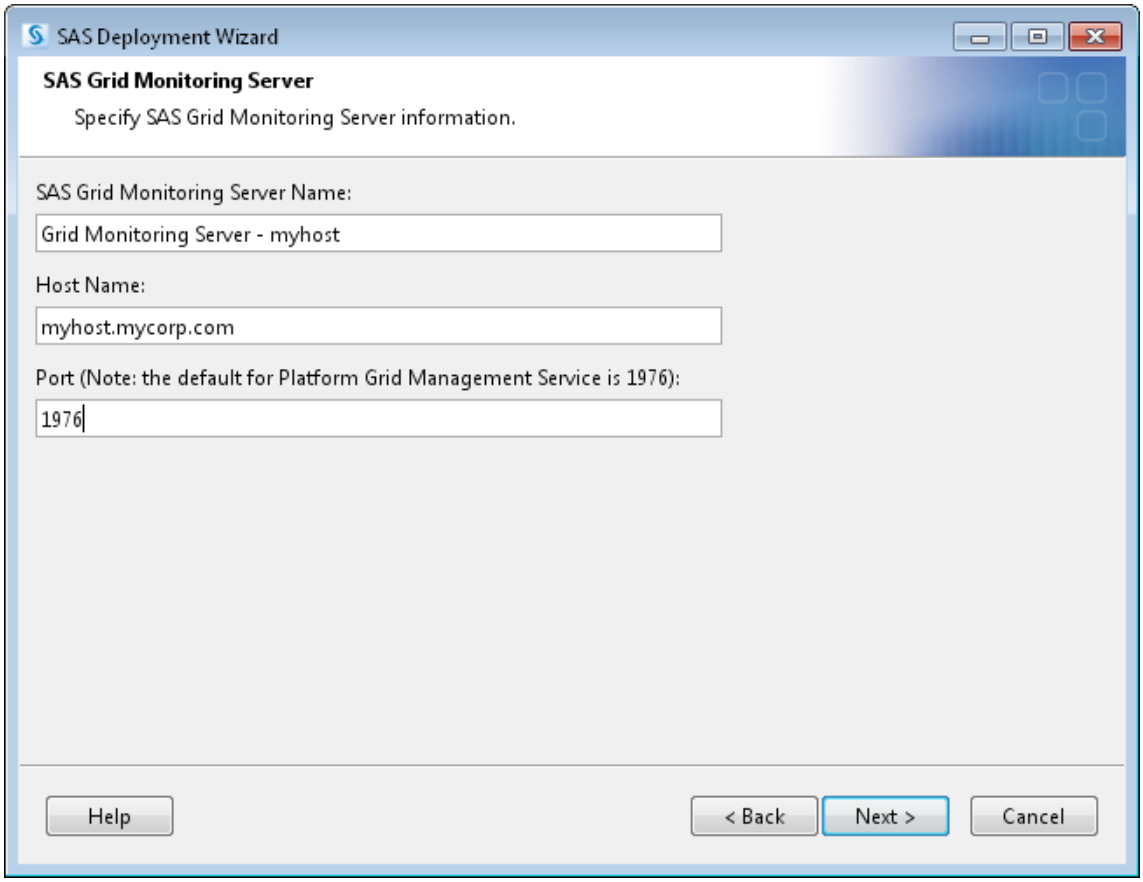
2 The Grid Control Server: Job Information page enables you to specify how jobs run on the grid. Specify the command used to start the server session on the grid, workload values, and additional options for the grid. The directory in the **Grid Shared Directory Path** field is used by grid programs (such as the SAS Grid Manager Client Utility) to store information. The location must be accessible by all grid nodes, and all grid users must have Read and Write access to the directory. For information about the values used in these fields, see "Modifying SAS Logical Grid Server Definitions" on page 210.

Figure 6.3 Grid Control Server: Job Information Page



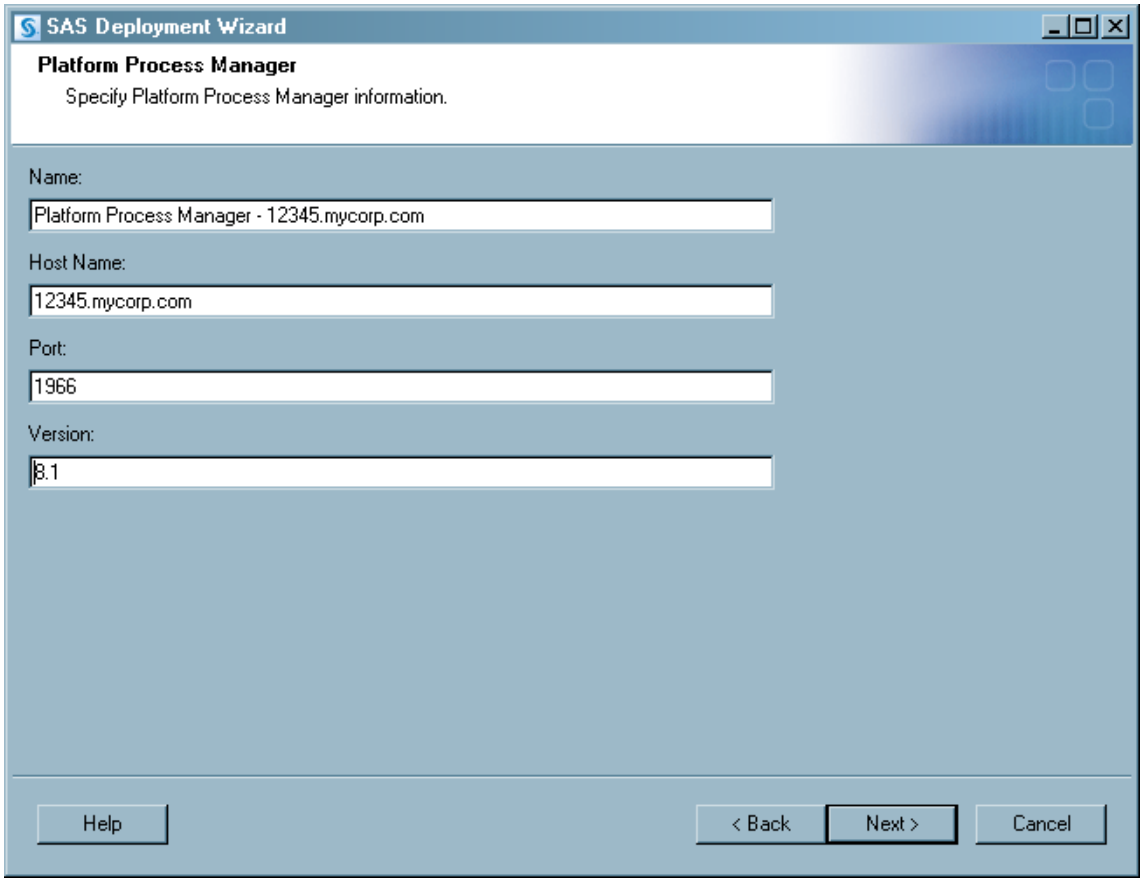
3 The SAS Grid Monitoring Server page enables you to specify the name, machine, and port for the grid monitoring server.

Figure 6.4 SAS Grid Monitoring Server Page



4 The Platform Process Manager page enables you to specify the server on which you installed Platform Suite for SAS and the port that is used to connect to the server.

Figure 6.5 Platform Process Manager Page for Custom Install

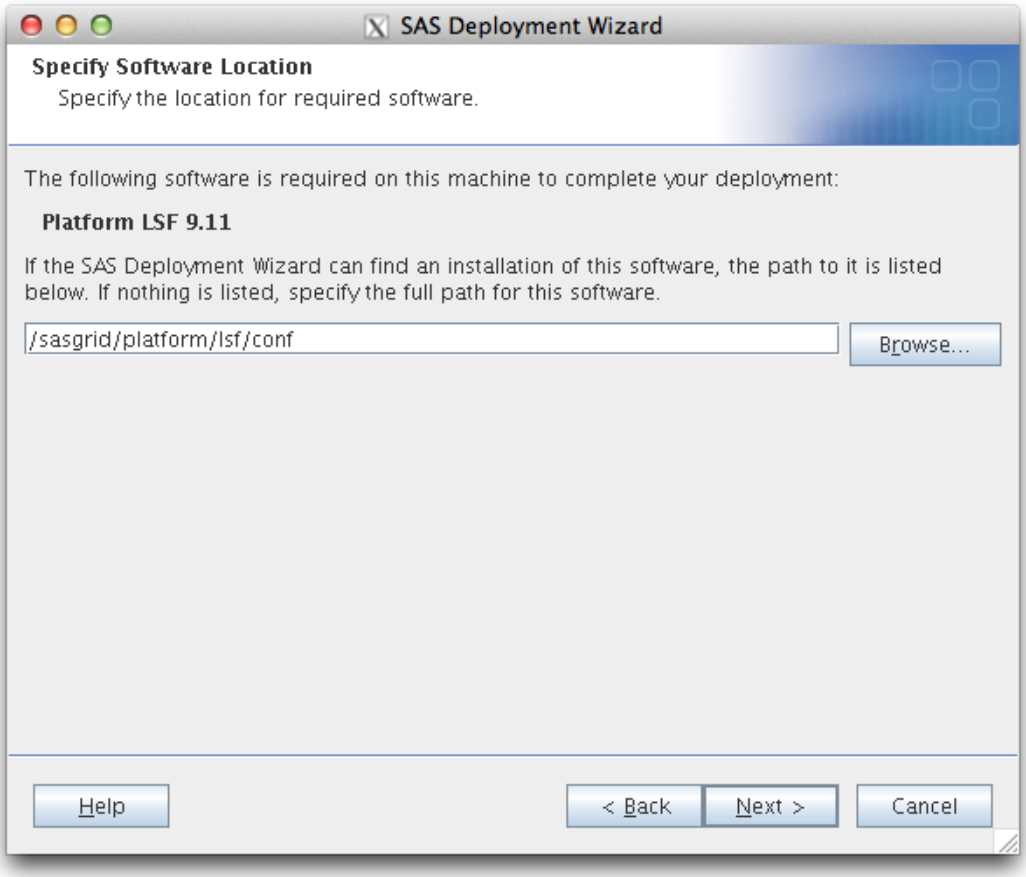


Configuring Platform Web Services

After you have configured the grid control server, you can use the SAS Deployment Wizard to configure Platform Web Services (PWS). The SAS Deployment Wizard requests the following information:

1 Use the Specify Software Location page to specify the location of your Platform LSF configuration file. The correct path is displayed by default if you correctly sourced the LSF profile script before you started the SAS Deployment Wizard.

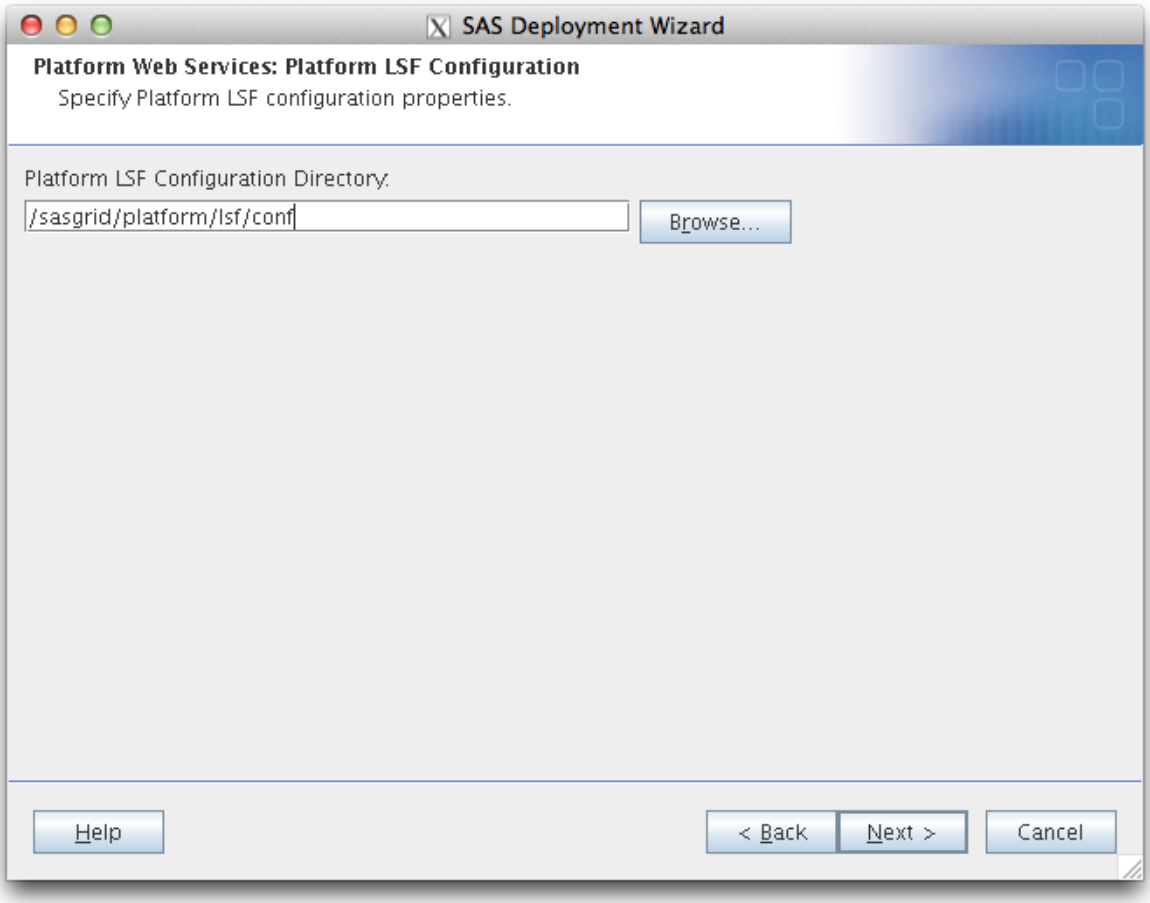
Figure 6.6 Specify LSF Location Page



2 If you specified that SAS Deployment Wizard should display custom prompts, you are prompted to specify a name for a SAS Web Application Server for PWS. This server must be dedicated to PWS. The default value of SASServer14 should not be changed.

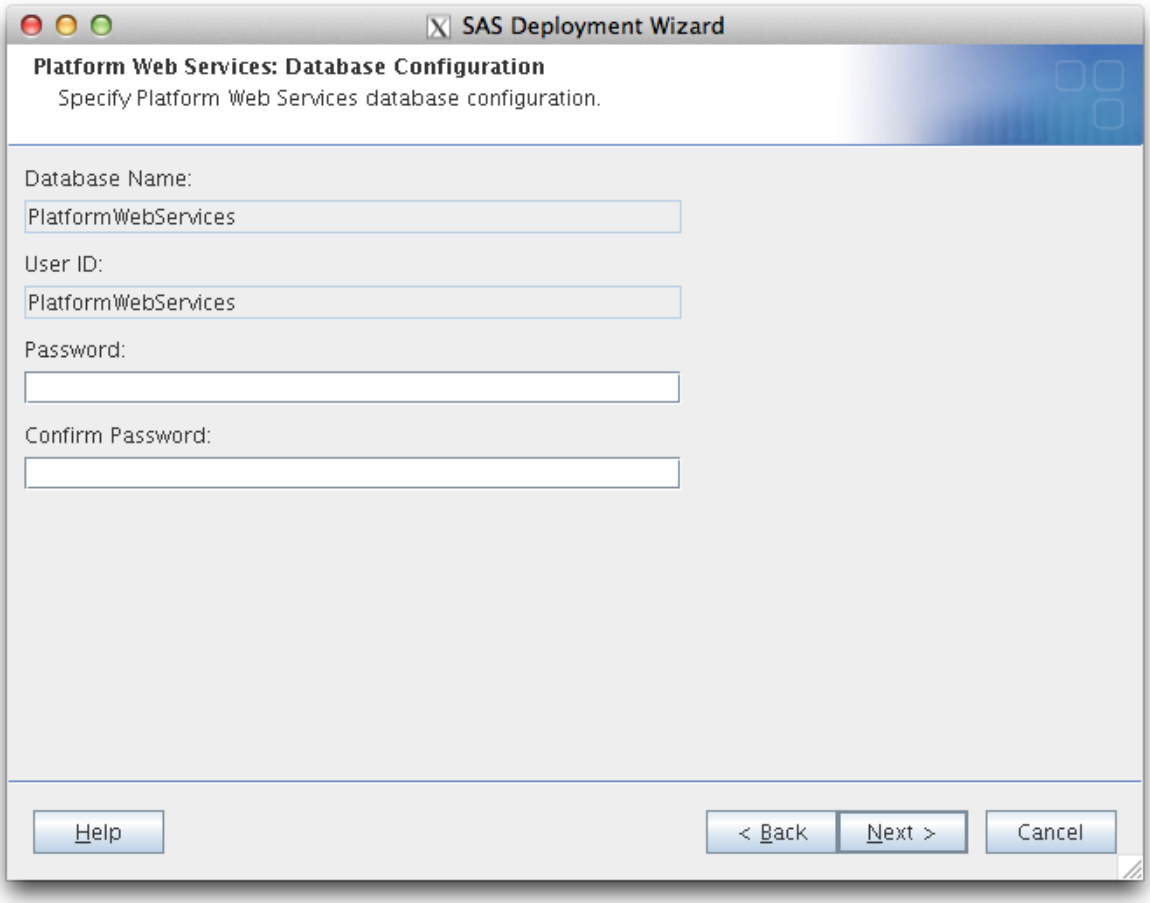
3 Use the Platform LSF Configuration page to specify the location of the Platform LSF configuration directory. The correct location should be specified by default.

Figure 6.7 LSF Configuration Directory Page



4 The Platform Web Services: Database Configuration page displays the Platform Web Services database and user ID, as specified in the Platform LSF configuration file. Specify and confirm the password for the user ID.

Figure 6.8 Platform Web Services Database Configuration Page



5 If you are installing on Windows, you must configure the SAS Web Application Server that runs PWS to run as the LSF Administrator.

   At a Windows prompt, type `services.msc` to open the Services window.

6 Stop the service for the SASServer14 SAS Web Application Server.

7 Modify the service to run as the LSF Administrator.

8 Restart the service.

Configuring the Grid Nodes

After you have installed and configured Platform Web Services, you can use the SAS Deployment Wizard to configure the grid nodes. The SAS Deployment Wizard installs and configures these components:

Table 6.2 Required Software Components for Grid Nodes

Installed SAS Software Components	Configured SAS Software Components
SAS Foundation (Base SAS, SAS/CONNECT)	SAS Grid Node Platform Suite for SAS

If more than one application server contains a logical grid server, you must choose which application server to use.

For information about the values required during a planned installation, see *SAS Intelligence Platform: Installation and Configuration Guide*.

Note: The configuration directory structure for each grid node must be the same as that of the grid control server.

Configuring Client Applications

After the grid nodes have been installed and configured, you can install and configure the software required for the client applications that will use the grid. The software required depends on the type of client application. Applications such as SAS Data Integration Studio that can submit jobs through a workspace server do not need to install anything other than the client application. Applications such as Base SAS that submit jobs to the grid must also install Platform Suite for SAS in order to send jobs to the grid. In releases prior to SAS 9.4M6, when you install SAS Management Console, you must also install the SAS Grid Manager plug-in. SAS Management Console is used to monitor and control the grid.

Note: Starting in SAS 9.4M7, Grid Management Services (GMS) is removed from Platform Suite for SAS. GMS version 8.01, which was delivered with SAS 9.4M6, is not compatible with Platform Suite for SAS 11.1 (including Process Manager 10.2 and LSF 10.16), which was also delivered with SAS 9.4M6. Therefore, GMS and the Grid Manager plug-in for SAS Management Console are in Limited Support. See SAS Technical Support Services and Policies for more information about Limited Support.

Modifying Grid Monitoring Server Definitions

The initial configuration of the grid monitoring server is performed by the SAS Deployment Wizard. However, a SAS grid administrator might need to modify the existing grid metadata or add new grid metadata definitions.

A SAS administrator performs these steps to specify or modify the required and optional metadata properties for the Grid Monitoring Server:

1. In SAS Management Console, open the metadata repository that contains the metadata for the SAS Grid Server.
2. In the navigation tree, select **Server Manager**.
3. Find the metadata object for the Grid Monitoring Server.
4. Right-click the metadata object for the Grid Monitoring Server, and then select **Properties**.
5. In the Properties window for the Grid Monitoring Server, click the **Options** tab.
6. Here are the fields on the **Options** tab:

   Provider
   the grid middleware provider. This value is used to communicate with the grid control server.

   Module Name
   specifies the shared library name or the class name of the support plug-in for Platform Suite for SAS. Leave this field blank unless directed otherwise by SAS Technical Support.

   Options
   the options needed by the grid monitoring server to connect to the grid server.

   RTM Host Name (property is relevant only for installations prior to SAS 9.4M6 that used Platform RTM)
   specifies the URL for the Platform RTM for SAS host.

   > **IMPORTANT** Platform RTM is not part of Platform Suite for SAS. The version of RTM that was available as a download for use with SAS Grid Manager is in Limited Support. See SAS Technical Support Services and Policies for more information about Limited Support. Contact SAS Technical Support for information about RTM.

7. After you complete the field entries, click **OK** to save the changes and close the Grid Monitoring Server Properties window.

8 In the display area (right side) on SAS Management Console, right-click the **Connection** object for the Grid Monitoring Server, and then select **Properties**.

9 In the Properties window for the Grid Monitoring Server Connection, click the **Options** tab. Here are the fields on this tab:

Authentication Domain (property is relevant only for installations prior to SAS 9.4M6)
: the authentication domain that is used for connections to the server. This value is the authentication domain of the machine that Grid Management Services (GMS) is running on.

Note: Starting in SAS 9.4M6, GMS is in Limited Support. See SAS Technical Support Services and Policies for more information about Limited Support.

Host Name
: the network address of the grid control server.

Port
: the port that is used to connect to the grid control server. The default value is 1976 for Platform Grid Management Service.

10 After you complete the entries, click **OK** to save the changes and close the Grid Monitoring Server Connection Properties window.

Installing and Configuring SAS Environment Manager in a Grid Environment

If you are installing SAS Environment Manager in a grid environment where more than one grid machine shares a configuration directory, you must perform additional configuration steps so that the SAS Environment Manager agents point to the correct machine. To run the configuration process, use the script deploy-ev-agents.sh, which you can access from SAS Environment Manager in a Grid Environment: deploy script.

If you are using the SAS Grid Manager agent plug-in or module for SAS Environment Manager, you must use the SAS Deployment Wizard to install and configure Platform Web Services on the middle-tier server.

The SAS Deployment Wizard prompts you for the following information:

- Context root for Platform Web Services
- Platform LSF configuration directory
- The name of the Platform Web Services database, as well as a user ID and password for connections to the database (the database name and user ID are grayed out and cannot be changed)

Using Kerberos Authentication in a Grid Environment

Setting Up Kerberos Authentication

If your deployment is using Kerberos authentication with grid-launched workspace servers, you must change several configuration parameters. These parameters enable ticket-granting ticket (TGT) forwarding, which makes Kerberos tickets available in order for SAS processes to authenticate to external resources. The Kerberos key distribution center (KDC) and the client configuration must allow forwarding. Follow these steps:

1. In the **lsf.conf** file, specify the parameter LSB_KRB_TGT_FWD=Y. This parameter turns on TGT forwarding.

2. Ensure that the krb5 libs (**libkrb5.so**, **libcom_err.so**, **libk5crypto.so**, and **libkrb5support.so**) are in the default directories. On 32-bit platforms, the directories are **/lib**, **/usr/lib**, and **/usr/local/lib**. On 64-bit platforms, the directories are **/lib64**, **/usr/lib64**, and **/usr/local/lib64**. If the libs are not in these directories, specify the directories in the LSB_KRB_LIB_PATH= parameter. You can specify multiple paths, separated by blanks, commas, or semicolons.

   If your libkrb5 libs are named `.so.`*`version`* rather than `.so`, create an unversioned symlink for your versioned libraries. This is an example of these symlinks.

   ```
   libkrb5.so -> ../../lib/libkrb5.so.3.2
   libcom_err.so -> ../../lib/libcom_err.so.1
   libk5crypto.so -> ../../lib/libk5crypto.so.3.1libkrb5support.so ->
   ../../lib/libkrb5support.so.1
   ```

3. If you are on AIX, specify the parameter LSB_KRB_AIX_LOAD_IBM_NAS=Y. This parameter looks for the file libkrb5.a in a location specified on the LSB_KRB_LIB_PATH= parameter and loads the symbols from this file.

4. In the **lsf.conf** file, specify any of these optional parameters:

   LSB_KRB_CHECK_INTERVAL
   : specifies the time interval for TGT checking. The default value is 15 minutes.

   LSB_KRB_RENEW_MARGIN
   : specifies the amount of time that elapses before the TGT is renewed. The default value is one hour.

   LSB_KRB_TGT_DIR
   : specifies where to store the TGT on the execution host. The default value is **/tmp**.

5. Restart LSF.

6 If you are using grid-launched workspace servers, and users are connecting to the servers with a network authentication provider, modify the **level_env_usermods.sh** file to add the line:

```
export SAS_GRID_USE_KERBEROS=1
```

If you are using end-to-end Kerberos authentication in an environment where all clients and all servers are on Windows, do not make this change.

Note: If you are using the full version of LSF, instead of Platform Suite for SAS, you must also add this line to the **level_env_usermods.sh** file:

```
export LSF_FULL_VERSION=9
```

Identifying Problems with Kerberos Authentication

If you are experiencing errors while setting up Kerberos authentication on your grid, you can set logging options to identify and isolate the problems. Follow these steps:

1 In the logging configuration file for the SAS Object Spawner (***SASCONFIGLEV*/ObjectSpawner/logconfig.xml**), locate the entry for the App.tk.tkegrid logger.

Change the `level value=` parameter to `Trace` or `Debug`.

2 In the same file, locate the entry for the Audit.Authentication logger.

Change the `level value=` parameter to `Trace` or `Debug`. Save and close the file.

3 In the LSF configuration file (***$LSFTOP*/conf/lsf.conf**), add these entries:

- LSB_LOG_MASK=LOG_DEBUG
- LSB_DEBUG_CMD="LC2_KRB"
- LSB_CMD_LOGDIR=*logging_directory*
- LSF_LOG_MASK=LOG_DEBUG
- LSF_DEBUG_CMD="LC2_KRB"
- LSF_CMD_LOGDIR=*logging_directory*

Save and close the file.

7

Managing a SAS Grid Manager for Platform Grid

Overview of Grid Management

Most organizations that use SAS consist of a variety of user categories. Each user category has its own needs and expectations. Some users in your environment might be running jobs that have a high priority. Other users might be running jobs

that require a large number of computing resources. A SAS grid environment must be able to account for all of these different needs, priorities, and workloads.

In order to manage this type of environment, you must be able to control when and where jobs can run on the grid. You can manage grid resources on a SAS Grid Manager for Platform grid using these strategies:

- Job slots. They enable you to control how many jobs can run concurrently on each machine on the grid. This enables you to tune the load that each machine on the grid can accept. For example, you can assign a higher number of job slots to higher-capacity machines, which specifies that those machines can process more jobs concurrently.
- Queues. They enable you to control when jobs can run and what computing resources are available to the jobs that are submitted to the queue. You can create queues based on factors such as job size or priority. You can also define job dispatch windows and run windows for each queue. When you submit a job to a particular queue, the queue settings determine when the job runs and what priority the job has compared to other jobs that have been submitted to the grid. You can also specify the number of job slots across the grid that a queue can use at any one time. By combining the job-slot specification on the hosts and queues, you can specify how work is distributed across the grid.
- Resources. They enable you to specify where jobs are run on the grid by specifying resource names on hosts and using matching resource names on jobs. The resource names are specified on machines on the grid to indicate what type of job each machine should run. When you submit jobs to the grid, you can specify resource names to specify which machines should be used to process the job.
- Grid options sets. They enable you to create sets of SAS options, required resources, and grid options. Each options set can then be mapped to a specified combination of a SAS application and a user or user group. For example, you might want to define policies for different applications running on the grid (such as SAS Data Integration Studio and batch SAS programs) as well as for different business units that are using those applications on the grid. After defining the grid, you can define grid options sets that contain the options needed for the different type of processing. Then, you can map the appropriate options sets to each of the user and application combinations that you need.

Modifying Configuration Files with Platform RTM for SAS

Note: This topic applies only to older installations that used Platform RTM for SAS. In SAS 9.4M3, a SAS Grid Manager Module for SAS Environment Manager was added. The Grid Manager Module for SAS Environment Manager serves the function previously provided by Platform RTM for SAS.

You can use Platform RTM for SAS to modify the configuration files that define queues and resources on a SAS Grid Manager for Platform grid. The Platform RTM for SAS download package contains documentation about performing this task.

However, if you use Platform RTM for SAS to change any configuration files, you cannot make any further changes to the files outside of Platform RTM for SAS. Changes in the configuration files are not synchronized with Platform RTM for SAS.

IMPORTANT Platform RTM is not part of Platform Suite for SAS. The version of RTM that was available as a download for use with SAS Grid Manager is in Limited Support. See SAS Technical Support Services and Policies for more information about Limited Support. Contact SAS Technical Support for information about RTM.

Note: Although you can use either the Grid Manager Module for SAS Environment Manager or Platform RTM for SAS to modify your LSF configuration, you cannot alternate between the two. There is no communication or synchronization between the Grid Manager Module and Platform RTM for SAS. For example, if you specify an LSF configuration as active in the Grid Manager Module and then use Platform RTM for SAS, Platform RTM for SAS does not know that the LSF configuration has been modified and consequently works with the wrong parameters and values.

Specifying Job Slots for Machines

Overview

Platform LSF uses job slots to specify the number of processes that are allowed to run concurrently on a machine. A machine cannot run more concurrent processes than it has job slots. The default number of job slots for a machine is the same as the number of processor cores in the machine.

However, you can configure hosts with fast processors to have more jobs slots than the number of cores by setting the MXJ value for the given host to a fixed number of job slots. This enables the more powerful host to execute more jobs concurrently to take advantage of the processor's speed.

Specifying Job Slots Using the SAS Grid Manager Module

To use the SAS Grid Manager module to specify the maximum number of job slots, follow these steps:

1. From the SAS Grid Manager main window, select **LSF configuration** in the left navigation area.

2 In the **LSF Configurations** table, select the entry for the configuration that you want to edit.

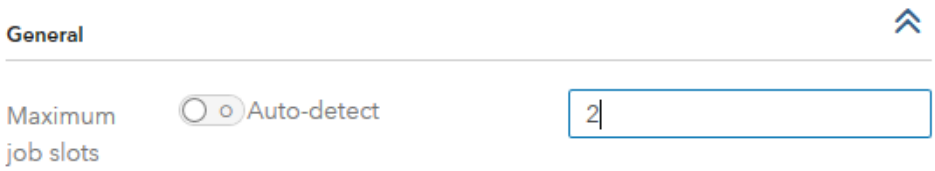
3 Select the **Batch hosts** tab. The table of defined batch hosts appears.

4 To specify the maximum number of job slots for all machines on the grid, expand the entry for the **default** host name. To specify the maximum number of job slots for a specific machine, expand the entry for that machine.

5 Expand the **General** area.

6 To specify that the number of job slots is equal to the number of cores, activate the **Auto-detect** switch. For example, a machine with 16 cores would have 16 job slots.

7 To specify the maximum number of job slots that is independent of the number of cores, turn off the **Auto-detect** switch and enter the number of job slots in the text field that appears beside the **Auto-detect** switch. For example, a value of 2 in this field results in two job slots on each machine regardless of how many cores are present.



8 Click ⊟ to save the definition.

9 Click ⁝≡ to return to the table of LSF configurations.

10 Select the check box for the configuration that you edited and click **Apply**.

Specifying Job Slots Manually

To change the number of job slots on a grid node by editing the `lsb.hosts` file, follow these steps:

1 Log on to the grid control server as the LSF Administrator (lsfadmin).

2 Open the file `lsb.hosts`, which is located in the directory *LSF-install-dir*/`conf/lsbatch/`*cluster-name*/`configdir`. This is the LSF batch configuration file. Locate the Host section of the file, which contains an entry for a `default` grid node.

```
Begin
Host
HOST_NAME MXJ    r1m     pg    ls    tmp  DISPATCH_WINDOW  #Keywords
default    !     ()      ()    ()     ()      ()            #Example
End Host
```

3 Edit this file to specify the maximum number of job slots for all nodes or for each node. There are two ways you can do this.

- To specify the total number of job slots per node, edit the line for the **default** node. Here is an example:

```
Begin Host
HOST_NAME MXJ    r1m     pg    ls    tmp  DISPATCH_WINDOW  #Keywords
default    !     ()      ()    ()     ()      ()            #Example
End Host
```

  The value ! represents one job slot per core for each node on the grid. You can replace this value with a number that specifies the maximum number of job slots on each node, regardless of the number of cores. For example, a value of ! on a machine with 16 cores results in 16 job slots, while a value of 2 on a machine with 16 cores results in just 2 job slots.

- To specify the total number of jobs slots per node, add a line for each node on the grid. Here is an example:

```
Begin
Host
HOST_NAME MXJ     r1m     pg    ls    tmp  DISPATCH_WINDOW  #Keywords
default     !     ()      ()    ()     ()      ()            #Example
D1234      16     ()      ()    ()     ()      ()            #Example
D1235      16     ()      ()    ()     ()      ()            #Example
D1236      16     ()      ()    ()     ()      ()            #Example
D1237      16     ()      ()    ()     ()      ()            #Example
D1238      16     ()      ()    ()     ()      ()            #Example
End Host
```

  Each line designates the concurrent execution of 16 jobs on each node.

4. Save and close the file.
5. Verify the LSF batch configuration file and reconfigure the LSF batch system by entering this command at the command prompt: **badmin reconfig**
6. For details about using this command, see the *IBM Spectrum LSF command reference* at https://www.ibm.com/support/knowledgecenter

Using Queues

Understanding Queues

When a job is submitted for processing on the grid, it is placed in a queue and is held until resources are available for the job. LSF processes the jobs in the queues based on parameters in the queue definitions that establish criteria such as which jobs are processed first, what hosts can process a job, and when a job can be processed. All jobs submitted to the same queue share the same scheduling and control policy. By using multiple queues, you can control the workflow of jobs that are processed on the grid.

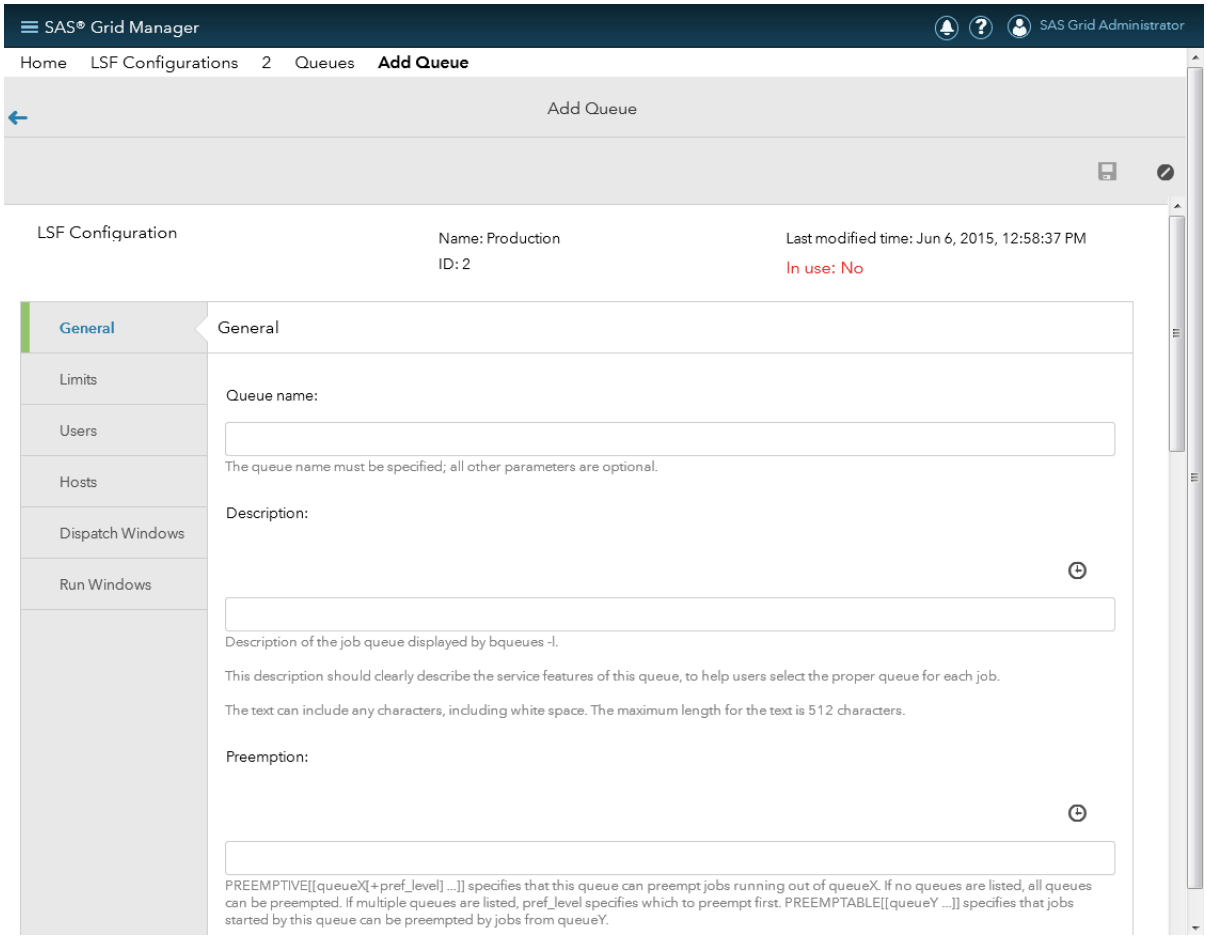
By default, SAS uses a queue named NORMAL. To use another queue that is already defined in the Grid Manager module or in the lsb.queues file, specify the

queue using a queue=*queue_name* option. You can specify this option in the metadata for the SAS logical grid server (in the **Grid Options** field), in the job options macro variable referenced in the GRDSVC_ENABLE statement, or in the **Grid Options** field of a grid options set. For information about specifying a queue in the logical grid server metadata, see "Modifying SAS Logical Grid Server Definitions" on page 210. For information about specifying a queue in a GRDSVC_ENABLE statement, see "GRDSVC_ENABLE Function" on page 261.

Configuring Queues Using the SAS Grid Manager Module

The SAS Grid Manager module for SAS Environment Manager enables you to define queues. Follow these steps to define a new queue:

1. From the SAS Grid Manager main window, select **LSF configuration** in the left navigation area.
2. In the **LSF Configurations** table, select the entry for the configuration that you want to edit.
3. Select the **Queues** tab. A list of defined queues appears.
4. Click + in the toolbar to define a new queue. The Add Queue window appears.



5 Specify values for the queue as described in the following topics.

6 When you finish defining the queue, click to save the definition.

7 Click to return to the table of LSF configurations.

8 Select the check box for the configuration that you edited and click **Apply**.

Configuring Queues Manually

Queues are defined in the lsb.queues file, which is located in the directory *LSF-install-dir*/`conf`/`lsbatch`/*cluster-name*/`configdir`. The file contains an entry for each defined queue. Each entry names and describes the queue and contains parameters that specify the queue's priority and the attributes associated with the queue. For a complete list of parameters allowed in the lsb.queues file, refer to *Platform LSF Reference*.

Using the Normal Queue

As installed, SAS Grid Manager for Platform uses a default queue called NORMAL. If you do not specify the use of a different queue, all jobs are routed to this queue and are processed with the same priority. Other queues enable you to use priorities to control the work on the queues.

In the SAS Grid Manager module, the values for the queue definition are the following:

Queue name field
normal

Priority field
30

Description field
default queue

The queue definition for the normal queue looks like the following:

```
Begin Queue
QUEUE_NAME = normal
PRIORITY = 30
DESCRIPTION = default queue
End Queue
```

Example: A High-Priority Queue

This example shows the existing queue for high priority jobs. Any jobs in the high-priority queue are sent to the grid for execution before jobs in the normal queue. The relative priorities are set by specifying a higher value for the PRIORITY attribute on the high-priority queue.

In the SAS Grid Manager module, specify the following values:

Queue name field
priority

Priority field
40

Description field
40

With the priority queue definition added to the normal queue definition, the manual queue definition file looks like the following:

```
Begin Queue
QUEUE_NAME = normal
PRIORITY = 30
DESCRIPTION = default queue
End Queue

Begin Queue
QUEUE_NAME = priority
PRIORITY = 40
DESCRIPTION = high priority users
End Queue
```

Example: A Night Queue

This example shows the existing queue for processing jobs (such as batch jobs) at night. The queue uses these features:

- The DISPATCH_WINDOW parameter specifies that jobs are sent to the grid for processing only between the hours of 6:00 PM and 7:30 AM.
- The RUN_WINDOW parameter specifies that jobs from this queue can run only between 6:00 PM and 8:00 AM. Any job that has not completed by 8:00 AM is suspended and resumed the next day at 6:00 PM.
- The HOSTS parameter specifies that all hosts on the grid except for host1 can run jobs from this queue. Because the queue uses the same priority as the normal queue, jobs from the high-priority queue are still dispatched first. Excluding host1 from the hosts that are available for the night queue leaves one host always available for processing jobs from other queues.

To define this queue in the SAS Grid Manager module, follow these steps:

1. Follow the general process for defining a queue. In the **General** tab, enter these values:

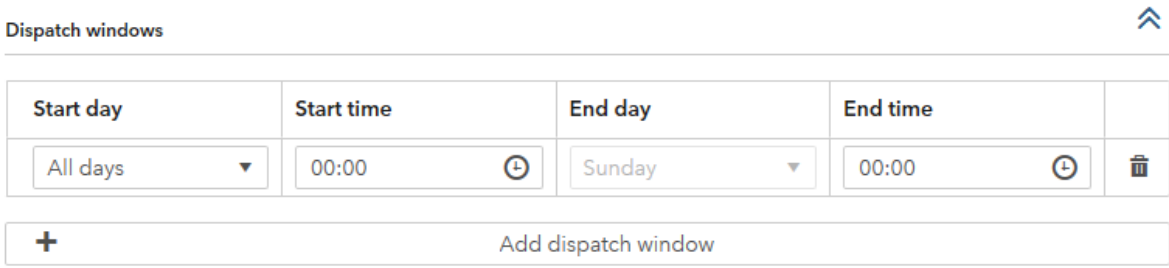
   Queue name field
   night

   Priority field
   30

   Description field
   night time batch jobs

2 Expand the **Dispatch Window** area to display the list of defined dispatch windows.

3 Click **Add dispatch window** to define a new dispatch window entry. An entry that is labeled 00:00 - 00:00 appears.



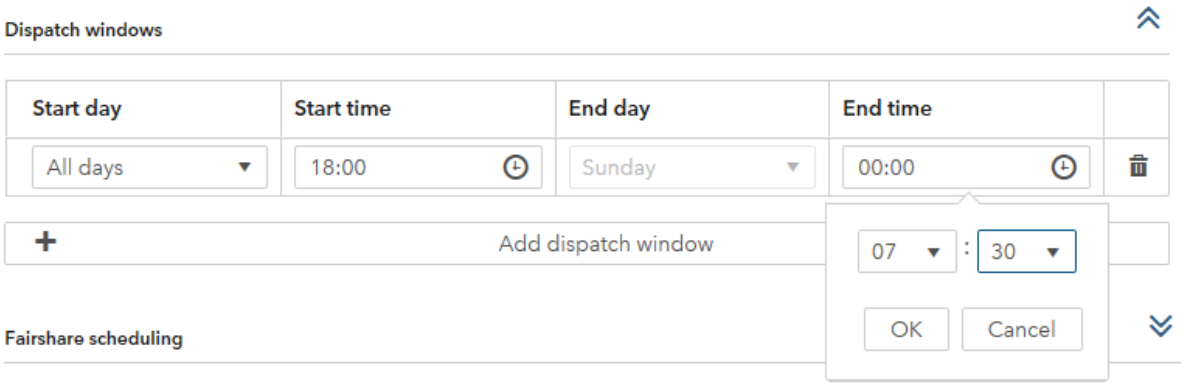
4 Specify the day of the week and the time at which the dispatch window starts and ends. For this example, select these values:

Start day
All days (the default value)

Start time
18:00

End day
Sunday (the default if **All days** is selected as the **Start day**)

End time
07:30



5 Expand the **Run Windows** area and follow the same process to specify a new run window with a start time of 18:00 and an end time of 08:00.

6 Expand the **General** area and specify the values all and ~host1 in the **Hosts** field.

7 Click ⊟ to save the configuration.

8 Click ⁝≡ to return to the table of LSF configurations.

Select the check box for the configuration that you edited and click **Apply**.

With the night queue definition added, the manual queue definition file looks like the following:

```
Begin Queue
```

```
QUEUE_NAME = normal
PRIORITY = 30
DESCRIPTION = default queue
End Queue

Begin Queue
QUEUE_NAME = priority
PRIORITY = 40
DESCRIPTION = high priority users
End Queue

Begin Queue
QUEUE_NAME = night
PRIORITY = 30
DISPATCH_WINDOW  = (18:00-07:30)
RUN_WINDOW = (18:00-08:00)
HOSTS = all ~host1
DESCRIPTION = night time batch jobs
End
Queue
```

Example: A Queue for Short Jobs

This example shows the existing queue for jobs that need to preempt longer-running jobs. The PREEMPTION parameter specifies which queues can be preempted as well as the queues that take precedence. Adding a value of PREEMPTABLE[short] to the normal queue specifies that jobs from the normal queue can be preempted by jobs from the short queue. Using a value of PREEMPTIVE[normal] to the short queue specifies that jobs from the short queue can preempt jobs from the normal queue. Using a value for PRIORITY on the short queue ensures that the jobs are dispatched before jobs from the normal queue. However, the jobs from the priority queue still take precedence.

To define these queues in the SAS Grid Manager module, follow these steps:

1. Edit the **normal** queue entry.
2. Expand the **General** area.
3. In the **Preemption** field, specify **PREEMPTIBLE[short]**. Save the queue entry.
4. Follow the general process for creating a new queue. Specify the following values:

   Name field
   short

   Priority field
   35

   Preemption field
   PREEMPTIVE[normal]

   Description field
   short duration jobs
5. Save the queue entry and apply the LSF configuration.

The manual queue definition file looks like the following:

```
Begin Queue
QUEUE_NAME = normal
PRIORITY = 30
PREEMPTION = PREEMPTABLE[short]
DESCRIPTION = default queue
End Queue

Begin Queue
QUEUE_NAME = priority
PRIORITY = 40
DESCRIPTION = high priority users
End Queue

Begin Queue
QUEUE_NAME = short
PRIORITY = 35
PREEMPTION = PREEMPTIVE[normal]
DESCRIPTION = short duration jobs
End Queue
```

Specifying Job Slot Limits on a Queue

A job slot is a position on a grid node that can accept a single unit of work or SAS process. Each host has a specified number of available job slots. By default, each host is configured with a single job slot for each core on the machine, so a multiple-core machine would have multiple job slots. For information about specifying job slots for a host, see *Platform LSF Reference*.

You can also use a queue definition to control the number of job slots on the grid or on an individual host that are used by the jobs from a queue. The QJOB_LIMIT parameter (the **Job slot limit** field) specifies the maximum number of job slots on the grid that can be used by jobs from the queue. The HJOB_LIMIT parameter (the **Per host job slot limit** field) specifies the maximum number of job slots on any single host that can be used by the queue. The following example specifies a limit of 60 job slots across the grid that can be used concurrently by the **normal** queue and a limit of 2 job slots on any host that can be used.

To define this queue in the SAS Grid Manager module, follow these steps:

1. Edit the **normal** queue definition entry.
2. Expand the **Limits** area. Specify these values:

   Job slot limit
   60

   Per host job slot limit
   2
3. Save the queue entry and apply the LSF configuration.

The entry in the manual definition file looks like the following:

```
Begin Queue
QUEUE_NAME = normal
```

```
PRIORITY = 30
DESCRIPTION = default queue
QJOB_LIMIT = 60
HJOB_LIMIT = 2
End Queue
```

Working with Mismatches between Queues and Workspace Servers

Defining a queue that specifies fewer hosts than are available for processing can cause an error. For example, you might want to have jobs in a certain queue to be processed only by the machine host2, so the queue definition would contain the line `HOSTS = host2`. You might also have a grid-launched logical workspace server definition that includes workspace servers for host1, host2, and host3. When the object spawner receives a request to start a workspace server, it then submits a job to Platform LSF to start the workspace server. Because the logical workspace server definition includes more hosts (host1, host2, host3) than the queue (host2), an error message results.

To accommodate this type of environment, add the line `ENABLE_HOST_INTERSECTION=Y` to the lsb.params file.

Alternatively, you can use the SAS Grid Manager module. Expand the **Additional attributes** area and click **Add additional attribute**. Specify ENABLE_HOST_INTERSECTION in the **Name** field and Y in the **Value** field.

Defining and Specifying Resources

Overview

Defining resources enables you to specify where jobs are run on the grid. You can specify resource names on grid nodes, and then specify those same resource names on jobs that are sent to the grid. The resource names that are specified on grid machines indicate a specific program or resource that the grid job might need (for example, a machine that has SAS/STAT installed on it or a machine that has access to a Teradata machine), so you can direct specific types of work to the nodes that are most appropriate for processing them.

By default, when a job is sent to the grid, the name of the SAS Application Server is sent as a resource name along with the job. You can further specify the type of machine used to run a job by specifying the WORKLOAD= parameter on the GRDSVC_ENABLE call.

For example, assume that you have installed and configured a grid that uses the application server name of SASApp. You now want to specify that SAS Data

Integration Studio jobs should run on certain machines on the grid. To make this happen, follow these steps:

1. Create a resource name of **DI** for SAS Data Integration Studio jobs. (DI is only an example; you can use any user-defined string.)
2. From the SAS Grid Manager main window, select **LSF configuration** in the left navigation area.
3. In the **LSF Configurations** table, select the entry for the configuration that you want to edit.
4. Select the **Cluster hosts** tab. Expand the entry for a host that you want to use for processing SAS Data Integration Studio jobs.
5. Expand the **Resources** area.
6. Enter the resource names **DI** and **SASApp** in the **Resources** field. Press Enter after each entry.
7. Save and apply the LSF definition.
8. Add the value **DI** to the **Required Resources** field for a grid options set definition that also specifies SAS Data Integration Studio in the **SAS Application** field.
9. When you send a job to the grid using SAS Data Integration Studio, the job is automatically sent to the DI workload, which sends the job to one of the machines with SASApp as a resource name and DI as a resource name. If there are no grid servers with resource names that match the value on the job, the job is not processed.

   If you did not use a grid options set, you would have to manually choose the workload named DI in the Loop Properties window in SAS Data Integration Studio.

Defining Resource Names Using Addresource

SAS Grid Manager for Platform provides the **`addresource`** command to define hosts and resources. To use this command to specify resource names, follow these steps:

1. Log on to the grid control machine as the LSF administrator.
2. If you are on UNIX, source the LSF profile file and verify that the PATH is correct.
3. Issue the command `addresource -r resource_name -m machine_name`. If the *machine_name* contains spaces, you must change the spaces to underscores.

   For example, the command `addresource -r DI -m D1234` assigns the resource name DI to the machine D1234.
4. Run the LSF commands to reconfigure the grid to recognize the new resources.

Specifying Resource Names Using GRDSVC_ENABLE

You can use the GRDSVC_ENABLE function to specify resource names for jobs that run on the grid. Use the SERVER= option to specify a resource to use if the specified grid server has specified a required resource. For more information, see “GRDSVC_ENABLE Function” on page 261.

Specifying Resource Names Using the SAS Grid Manager Client Utility

If you specify required resources in the metadata definition for a SAS Application Server, you can use the SAS Grid Manager Client Utility to specify that those resources are used. Specify the -GRIDAPPSERVER option to specify the SAS Application server to use. For more information, see “SASGSUB Command: Submitting a SAS Program in Batch Mode” on page 282.

Specifying Resource Names in SAS Data Integration Studio

In order to specify the resource name for SAS Data Integration Studio jobs, you must complete these tasks:

- Add the resource name as an allowed value for the logical grid server to which you send jobs.
- Specify the workload that corresponds to the resource name in the loop transformation properties.

To add the resource name to the logical grid server metadata's Workload values, see “Modifying SAS Logical Grid Server Definitions” on page 210.

To specify the workload value in SAS Data Integration Studio, follow these steps:

1. On the SAS Data Integration Studio menu bar, select **Tools** ⇨ **Options**, and then select the **SAS Server** tab on the Options dialog box.
2. Select the SAS grid server in the **Server** field.
3. Select the workload to use for the submitted jobs in the **Grid workload specification** field.

You can also specify the resource name through a grid options set, either on GRDSVC_ENABLE or by using the GRIDOPTSET parameter on the SAS Grid Manager Client Utility.

8

Using Grid Management Applications With SAS Grid Manager for Platform

Using SAS Environment Manager

Overview

If your grid uses Platform Suite for SAS, SAS Environment Manager provides an agent plug-in and a management module that enables you to monitor and manage a SAS grid cluster. The plug-in and module have different purposes, however:

SAS Grid Manager for Platform Agent Plug-in
: uses continuously collected metric data to monitor the performance of the grid, grid servers, and grid queues, to graph changing metric data, and to generate alerts.

SAS Grid Manager for Platform module
: configures and controls grid resources, views current performance data, and configures grid resources and high availability applications.

> **IMPORTANT** Platform RTM is not part of Platform Suite for SAS. The version of RTM that was available as a download for use with SAS Grid Manager is in Limited Support. See SAS Technical Support Services and Policies for more information about Limited Support. Contact SAS Technical Support for information about RTM.

See Installing and Configuring SAS Environment Manager in a SAS Grid Environment with a Shared Configuration Directory for information about installing and configuring SAS Environment Manager so that it can monitor grid resources.

Configuring Platform Web Services

You must install and configure Platform Web Services in order to use both the SAS Grid Manager for Platform plug-in and module. Use the SAS Deployment Wizard to install Platform Web Services. See "Installing and Configuring SAS Environment Manager in a Grid Environment" on page 110 for more information.

Platform Web Services must run under the account of the LSF Primary Administrator user. If the install account is not an administrator account, you can make the installer account an administrator account, or, if you are using Windows, you can change the account under which the service is running. The best practice is to specify the installer account as the LSF Primary Administrator.

If you are running on UNIX or Windows, follow these steps to make the installer account an LSF Primary Administrator account:

1. Edit the `lsf.cluster.`*`cluster_name`* file.
2. Locate the line `Begin ClusterAdmins` in the file.

3 In the next line, `Administrators=admin1admin2`, add the ID of the installer account. Separate IDs with a space.

4 Save and close the file.

5 Run the command `lsadmin reconfig`.

6 Run the command `badmin mbdrestart`.

If you are running on Windows, follow these steps to change the account under which the service is running:

1 In the Windows Services window, click the entry for the service **SASServer14_1** and select **Stop** from the pop-up menu.

2 In the dialog box for the SASServer14_1 service, click the **Log On** tab.

3 Select the **This account** radio button and specify the user ID and password of the LSF Administrator user. Click **OK** to close the dialog box.

4 Click the entry for the service and select **Start** from the pop-up menu.

Removing the SAS Grid Manager HQU Plug-in

In SAS 9.4M2, the SAS Grid Manager HQU plug-in was added to SAS Environment Manager. In SAS 9.4M3, this plug-in was replaced by the SAS Grid Manager module for SAS Environment Manager. Although leaving the old plug-in installed will not cause errors, it is recommended that it be removed.

To remove the SAS Grid Manager HQU plug-in, delete the folder ***SASconfig/Levelroot*/Web/SASEnvironmentManager/server-5.8.0-EE/hq-engine/hq-server/webapps/ROOT/hqu/GridManagerPlugin**

Using the SAS Grid Manager for Platform Agent Plug-in for SAS Environment Manager

Overview

The SAS Grid Manager for Platform agent plug-in for SAS Environment Manager provides metric data in a SAS Grid Manager for Platform environment for the grid cluster, individual grid hosts, and grid queues, and uses that data to perform these functions:

- display the current state of grid resources

- graph the data over time, providing a historical view and enabling you to see how the data changes
- create alerts that notify you whenever a selected measurement reaches a selected state

Configuring the SAS Grid Manager for Platform Plug-in

Before the SAS Grid Manager for Platform agent plug-in can discover the grid resources, it must be configured. Follow these steps to configure the plug-in:

1. Log on to SAS Environment Manager with an administrative user ID.
2. Select **Resources** ⇨ **Browse** ⇨ **Servers**. The **SAS Grid Manager for Platform** entry in the table contains a gray question mark icon in the **Availability** column, which indicates that it has not been configured. Select the **SAS Grid Manager for Platform** entry in the table to display the Resource Details page.
3. Select **Tools Menu** ⇨ **Configure Server** to display the Configuration Properties page.
4. Specify the user ID and password for the LSF Administrator user and click **OK**.

Monitoring SAS Grid Manager for Platform Grid Resources

Discovering SAS Grid Manager for Platform Grid Resources

After the plug-in is configured, SAS Environment Manager will automatically discover the resources associated with the SAS Grid Manager for Platform grid. By default, the process to discover new resources runs every 24 hours. Whether the plug-in is discovering resources for the first time, or it is discovering new or changed nodes on the grid, up to 24 hours could pass before the resources are discovered.

If you want to discover resources sooner, you can restart the SAS Environment Manager agent, which causes the discovery process to start immediately. If you want to change the interval at which new or changed resources are discovered, modify the file ***SASConfig*/Lev1/Web/SASEnvironmentManager/agent-5.8.0-EE/conf/agent.properties**. Modify the parameter `autoinventory.runtimeScan.interval.millis=`*`interval`*, where *`interval`* is the autodiscovery interval in milliseconds. The default value is `86400000`. Restart the SAS Environment Manager agent after you change the value of the parameter.

After the resources are discovered, SAS Environment Manager starts collecting metric data that you can use to monitor the health and operation of your SAS grid. You can view data for the entire grid and for individual grid nodes.

Viewing Availability and Metrics for a SAS Grid Manager for Platform Grid

To view availability and metrics for a SAS Grid Manager for Platform grid as a whole, follow these steps:

1. In SAS Environment Manager, select **Resources** ⇨ **Browse** and select **Servers**. A table of server resources appears.
2. Locate and select the **SAS Grid Manager for Platform** entry in the table.

   The details page displays availability and metric data for the SAS Grid Manager for Platform cluster. See "Using the Details Page for a Grid Resource" on page 133.

   Note: In a multi-resource environment, use the metric charts for individual resources rather than for the entire grid. The charts for the entire grid are an aggregate of the metrics for all the individual nodes, and can be misleading.

Here are the metrics that are collected for the grid cluster:

- Availability
- CPU Utilization
- Jobs Complete
- Jobs Pending
- Jobs Running
- Jobs Suspended
- Load Average (processes per 15 minutes)
- Load Average (processes per 15 seconds)
- Load Average (processes per one minute)
- Max Job Slots Available
- Max Physical Memory
- Max Swap Memory
- Number of Cores
- Physical Memory
- Swap Memory
- System IO
- System Paging

Viewing Availability and Metrics for a SAS Grid Manager for Platform Grid Node

To view availability and metrics for individual nodes on a SAS Grid Manager for Platform grid, follow these steps:

1. In SAS Environment Manager, select **Resources** ⇨ **Browse** and select **Services**. A table of services appears.
2. Locate and select a **SAS Grid Manager Host** entry in the table. The table contains a separate entry for each host that you have defined.

   The details page displays availability and metric data for the selected SAS Grid Manager for Platform host. See "Using the Details Page for a Grid Resource" on page 133.

These are the metrics that are collected for each host:

- Availability
- CPU Factor
- CPU Utilization
- Interactive Idle Time
- Is Master
- Jobs Complete
- Jobs Running
- Jobs System Suspended
- Jobs User Suspended
- Load Average (processes per 15 minutes)
- Load Average (processes per 15 seconds)
- Load Average (processes per minute)
- Login Sessions
- Max Job Slots Available
- Max Physical Memory
- Max Swap Memory
- Number of CPUs
- Number of Cores per Physical Processor
- Number of Local Disks
- Number of Physical Processors
- Number of Threads per Processor
- Physical Memory
- Swap Memory
- System IO

- System Paging

Viewing Availability and Metrics for SAS Grid Manager for Platform Grid Queues

To view availability and metrics for SAS Grid Manager for Platform grid queues, follow these steps:

1. In SAS Environment Manager, select **Resources** ⇨ **Browse** and select **Services**. A table of services appears.
2. Locate and select a **SAS Grid Manager Queue** entry in the table. The table contains a separate entry for each queue that you have defined.

   The details page displays availability and metric data for the selected SAS Grid Manager for Platform queue. See "Using the Details Page for a Grid Resource" on page 133.

Here are the metrics that are collected for each queue:

- Availability
- Jobs Complete
- Jobs Done
- Jobs Exited
- Jobs Pending
- Jobs Running
- Jobs System Suspended
- Jobs User Suspended
- NJOBS (total number of jobs)
- Pending Times

Using the Details Page for a Grid Resource

The details page for a grid resource displays an availability indicator and graphs for selected metrics for the selected resource. You can also view a table of all collected metrics for the resource.

The **Monitor** tab on the details page for a grid resource contains an **Availability** bar, which enables you to check the availability of the grid or grid host at a glance. The bar displays a color-coded indicator that represents the availability during a time slice (the length of which depends on the display range that you select). The percentage of time that the resource was available is displayed at the end of the availability bar. The color codes are:

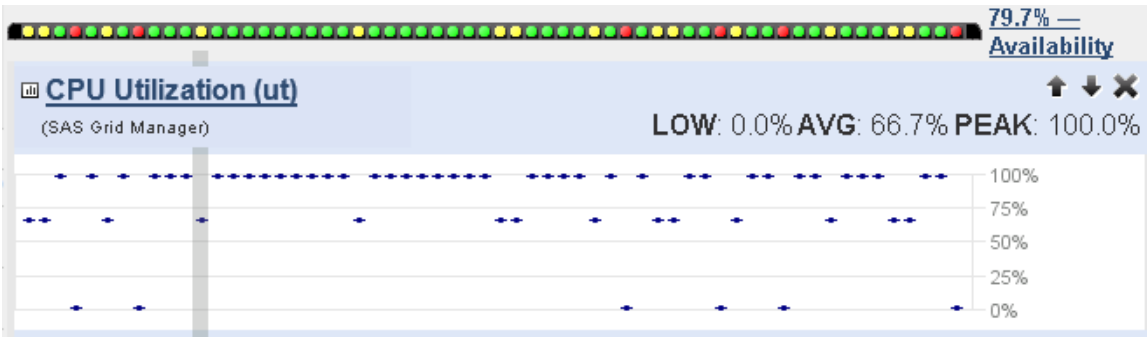
Green
: 100% availability

Yellow
: Partial availability (between 0% and 100%)

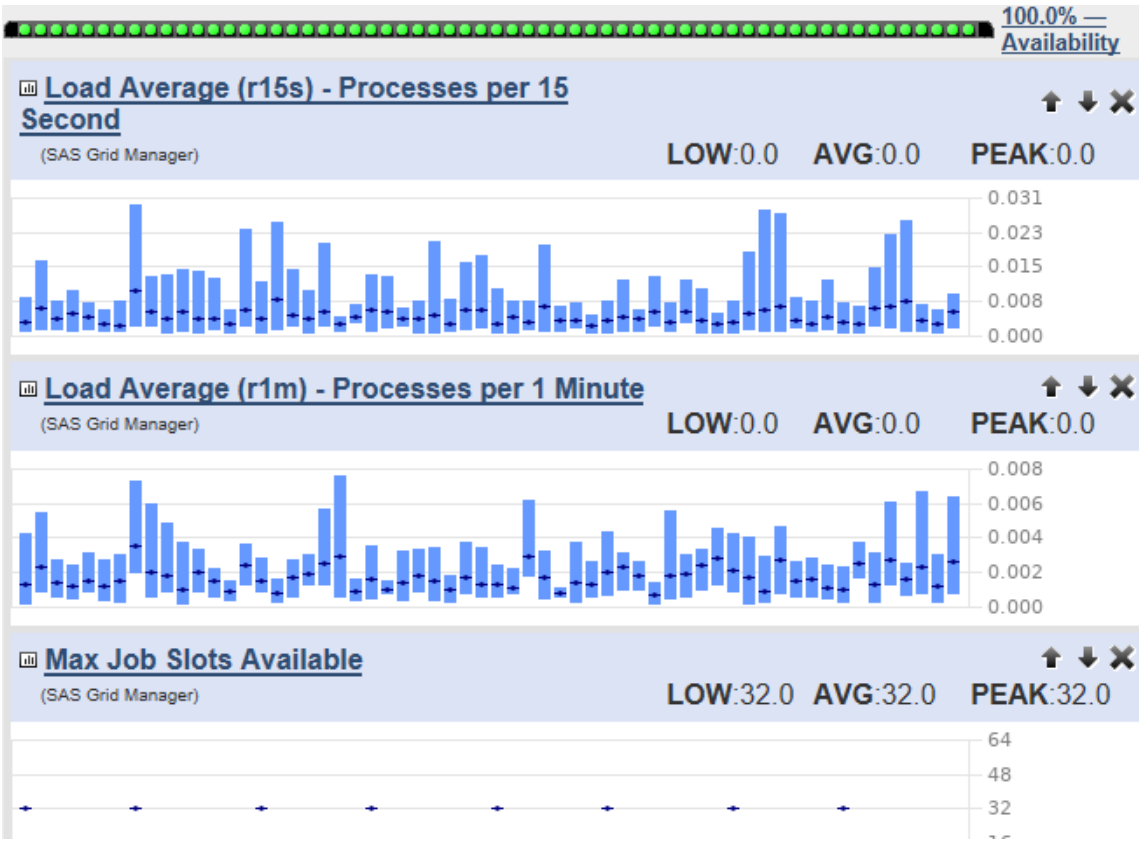
Red
: 0% availability

Click the percentage availability at the end of the availability bar to view a graph of the availability information.

Clicking on a dot in the availability bar highlights the time slice in the charts beneath the bar, which helps you diagnose availability problems.



Below the **Availability** bar on the **Monitor** tab is a set of charts that contain data for each collected metric over a selected time period. To change the time period displayed in the graphs, select a value in the **Metric Display Range** field. The default period is eight hours.



To view detailed metric data, click **Metric Data**. The details page changes to display the metric data in table form. By default, the table displays only the available data. To see all of the collected metric data, click the arrow next to **Show All Metrics**. By default, the metric data refreshes every two minutes. To change the refresh interval, click a value next to **Metrics Refresh**.

Indicators | Metric Data

Show All Metrics | Last Updated: 11/01/2018 05:00:28 PM | Metrics Refresh: 1 min | 2 min | 5 min | OFF

Availability	Alerts	OOB	LOW	AVG	PEAK	LAST	Collection Interval
Availability	0	0	-	100.0%	-		00:01:00
Utilization							
CPU Utilization (ut)	0	0	1.5%	2.1%	5.4%	2.1%	00:01:00
Interactive Idle Time (it)	0	0	1 day 17:37:36	1 day 21:37:26	2 days 01:37:01	2 days 01:37:01	01:00:00
Is Master	0	0	100.0%	100.0%	100.0%	100.0%	01:00:00
Jobs Complete	0	0	0.0	0.0	0.0	0.0	00:05:00
Jobs Running	0	0	0.0	0.0	0.0	0.0	00:01:00
Jobs System Suspended (ssusp)	0	0	0.0	0.0	0.0	0.0	00:05:00
Jobs User Suspended (ususp)	0	0	0.0	0.0	0.0	0.0	00:05:00
Load Average (r15m) - Processes per 15 Minute	0	0	0.0	0.0	0.0	0.0	00:05:00
Load Average (r15s) - Processes per 15 Second	0	0	0.0	0.0	0.0	0.0	00:01:00
Load Average (r1m) - Processes per 1 Minute	0	0	0.0	0.0	0.0	0.0	00:01:00
Login Sessions (ls)	0	0	1.0	1.0	1.0	1.0	00:05:00
Max Job Slots Available	0	0	16.0	16.0	16.0	16.0	01:00:00
Max Physical Memory (maxmem)	0	0	-	-	-	16.0 GB	01:00:00
Max Swap Memory (maxswp)	0	0	-	-	-	7.8 GB	01:00:00
Physical Memory (mem)	0	0	4.9 GB	5.0 GB	5.0 GB	5.0 GB	00:05:00
Swap Memory (swp)	0	0	7.5 GB	7.5 GB	7.5 GB	7.5 GB	00:05:00
System IO (io) - KB per sec	0	0	152.1	203.6	577.9	177.1	00:01:00
System Paging (pg) - Pages per sec	0	0	0.0	0.6	94.9	0.2	00:01:00

Viewing SAS Grid Manager for Platform Reports in the Data Mart

If you have enabled the SAS Environment Manager Service Architecture, you can also view reports produced by metric data from the grid that is collected in the SAS Environment Manager Data Mart. To view these reports, select **Analyze** ⇨ **Report Center**. In the Report Center, open the folders **Stored Processes** ⇨ **Products** ⇨ **SAS Environment Manager** ⇨ **Kits** ⇨ **SAS Grid Manager**. Use the prompts to produce reports that show the daily job completion totals per host and the job activity on the grid. See Initializing and Enabling the Service Architecture in *SAS® Environment Manager 2.5: User's Guide* for more information.

Creating Alerts

SAS Environment Manager enables you to create an alert for any of the metrics collected by the SAS Grid Manager for Platform agent plug-in. An alert is a user-defined type of event that indicates a critical condition in a selected resource. When an alert occurs, it must be acknowledged, and alerts are listed until they are marked as being fixed. You can define escalation schemes to identify the actions that happen if an alert is not fixed within a specified time.

For example, you can create an alert that occurs whenever availability of the grid cluster falls below a certain percentage, or when the number of available job slots falls below a certain threshold. See Working with Resource Alerts in *SAS® Environment Manager 2.5: User's Guide* for more information about alerts.

The plug-in includes these pre-defined alerts for these conditions:

- Availability of the grid (SAS Grid Manager resource) falls below 100%
- CPU utilization of the grid (SAS Grid Manager resource) goes above 95%. This alert occurs once for every 12 times the condition is met within one hour.
- CPU utilization of each grid host (SAS Grid Manager Host resource) goes above 95%. This alert occurs once for every 12 times the condition is met within one hour.

To create an alert, follow these steps:

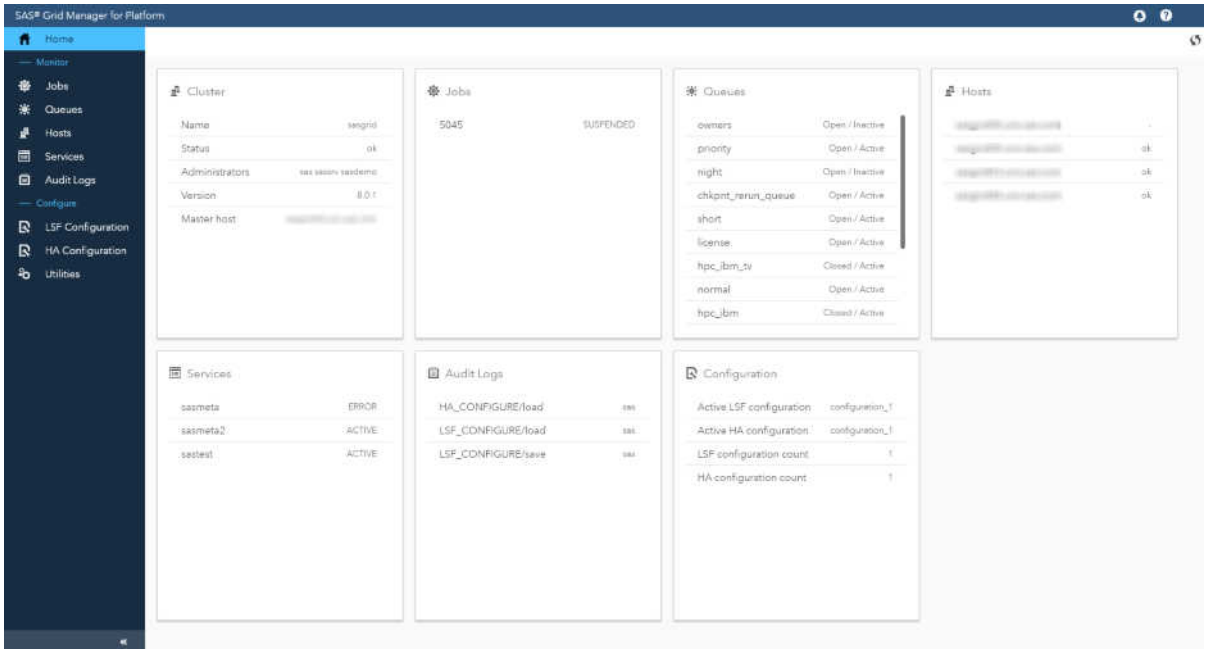
1. Select **Resources** ⇨ **Browse** and either **Servers** (for the grid cluster) or **Services** (for the grid nodes). Click the alert icon (which is in the group of three icons to the left of the entry in the table) for the resource. The Alerts page for the resource appears.
2. Click **New** to display the New Alert page.
3. In the New Alert page, specify the metric for which you want to create an alert and the condition for the metric that causes the alert. The page also enables you to define other criteria for the alert, such as how often the alert can occur. See Defining an Alert in *SAS® Environment Manager 2.5: User's Guide* for more information about defining an alert.

Using the SAS Grid Manager for Platform Module for SAS Environment Manager

Overview

The SAS Grid Manager for Platform module for SAS Environment Manager enables you to configure and perform actions on SAS Grid Manager for Platform grid resources and high availability applications, provides a snapshot of monitored data for grid resources, and enables you to update your LSF license file. Start the SAS Grid Manager for Platform module by selecting **SAS Grid Manager for Platform** from the SAS Environment Manager Administration side menu.

Note: You must be assigned to either the **Management Console: Advanced** role or the **Management Console: Content Management** role in SAS metadata in order to access the SAS Grid Manager for Platform module from the side menu.



The module displays a set of tiles that provide a view of the status of the grid. These tiles are displayed:

Cluster
displays the name of the LSF cluster, the cluster status, the IDs of cluster administrators, the version of Platform Suite for SAS, and the name of the master host. A cluster is a group of hosts that is organized and managed by an LSF administrator. Clusters are the basis for job sharing in Platform LSF.

Jobs
displays a list of jobs and the status of each job

Queues
displays a list of defined queues and the status (open or closed and active or inactive) of each queue

Hosts
displays a list of the hosts on the grid and the status of each host

Services
displays a list of defined services and the status of each service

Audit Logs
displays a list of recent records in the LSF audit log database

Configuration
displays the active LSF configuration, the active HA configuration, and the number of defined LSF and HA configuration instances

The left navigation menu enables you to access these functional areas:

Jobs
view detailed information about jobs on the grid and perform actions (kill, resume, requeue, and switch queue) on the jobs

Queues
view a list of queues, open or close queues, activate or reactivate queues, view detailed information about each queue

Hosts
view a list of hosts on the grid and detailed information about each host.

Services
view a list of services, stop, start, or migrate services, view detailed information about services and their use

Audit Logs
view a list of audit logs and detailed information about the logs

LSF configuration
view a list of defined LSF configurations, load, apply, or modify configurations

HA configuration
view a list of defined configurations for high availability applications, load, apply, or modify configurations

Managing Jobs

The SAS Grid Manager for Platform **Jobs** area enables users who are not LSF administrators to perform these functions:

- view the status of their own jobs
- suspend or terminate their own running jobs
- resume or terminate their own suspended jobs
- view detailed information and the history for their own jobs

The **Job** area enables LSF administrators to perform these functions:

- view the status of all jobs
- suspend or terminate all running jobs
- resume or terminate all suspended jobs
- terminate and requeue jobs
- switch an uncompleted job to a different queue
- change the position of jobs in the queue
- view detailed information and the history for each job

To view information about a job, follow these steps:

1. In the left navigation menu, select **Jobs**. The **Jobs** table appears, and it contains summary information for the jobs on the grid.
2. Click an entry in the table to view the **Job:*jobnumber*** window for the job.
3. The **Job Information** tab contains information about the job, including where it is (or has been) running, the name of the queue used, the job command, and the start time. The **Job History** tab contains information about the execution history of the job, including summaries of the job information, the time information (such as the run time, pending time, and suspended time), and the job contents (the events in job processing, such as submission, job start and end, and any error messages).

   The **General** tab contains a summary of information about the job, including the status, the name of the queue used, and the job command. The **Execution** tab contains detailed information about the job execution, including the host and directory where the job is (or has been) running, the input or output files used,

and the amount of swap memory used. The **Times** tab displays the start and end time for the job and the time-related metrics (such as CPU time, user time, and run time) for the job. The History tab displays a list of events for the job, such as being submitted to the grid, starting, and ending.

To perform actions on jobs in the SAS Grid Manager for Platform module, follow these steps:

1. In the left navigation menu, select **Jobs**.
2. In the table, select the check box for the job on which you want to perform an action. Select multiple check boxes to perform an action on more than one job.
3. Available actions appear at the top of the **Jobs** table. Choices for actions are:

   Cancel
   Stop execution of the selected job.

   Resume
   Resume processing of a suspended job.

   Requeue
   Terminate and requeue the job. The requeued job is placed after the jobs of the same priority. Do not use this function on an interactive job. You must be an LSF administrator to perform this function.

   Switch Queue
   Switch the job to the specified queue. The job can be switched only to a queue to which the job could have been submitted. You must be an LSF administrator to perform this function.

   Bottom
   Move the job to a position in the queue after the last job of the same priority. You must be an LSF administrator to perform this function.

   Run
   Force a pending job to run immediately on the specified host. You must be an LSF administrator to perform this function.

   Suspend
   Pause the execution of the selected job.

   Top
   Move the job to a position in the queue before the first job of the same priority. You must be an LSF administrator to perform this function.

4. Confirm the action when prompted.

Click the refresh icon to manually refresh the table.

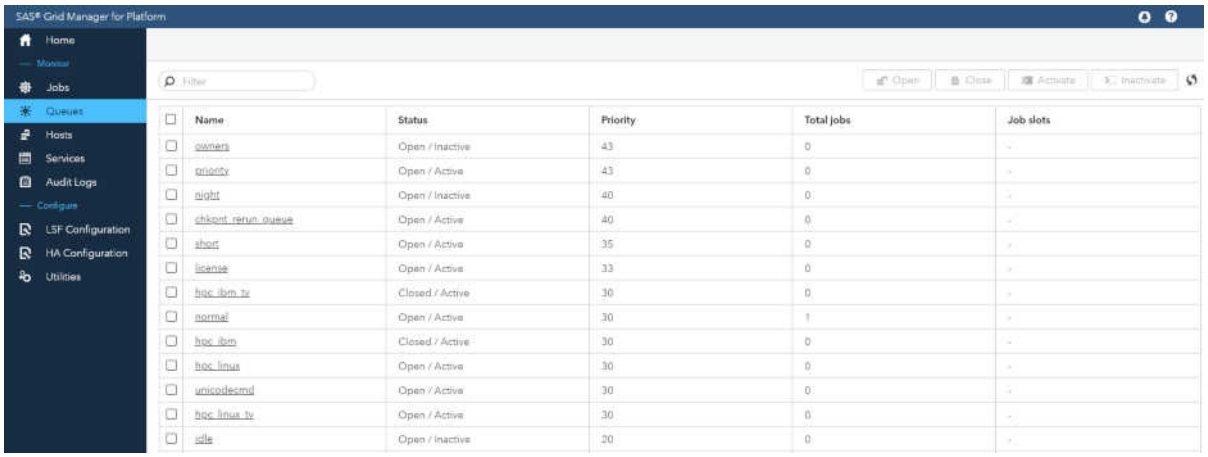
If you log on to the SAS Grid Manager for Platform module using a user ID that is defined as an LSF Administrator ID, you can manage any jobs that have been submitted to the grid. Users who are not LSF administrators can manage only their own jobs. See "Using the SAS Grid Manager for Platform Module as the LSF Administrator" on page 149 for more information.

Managing Queues

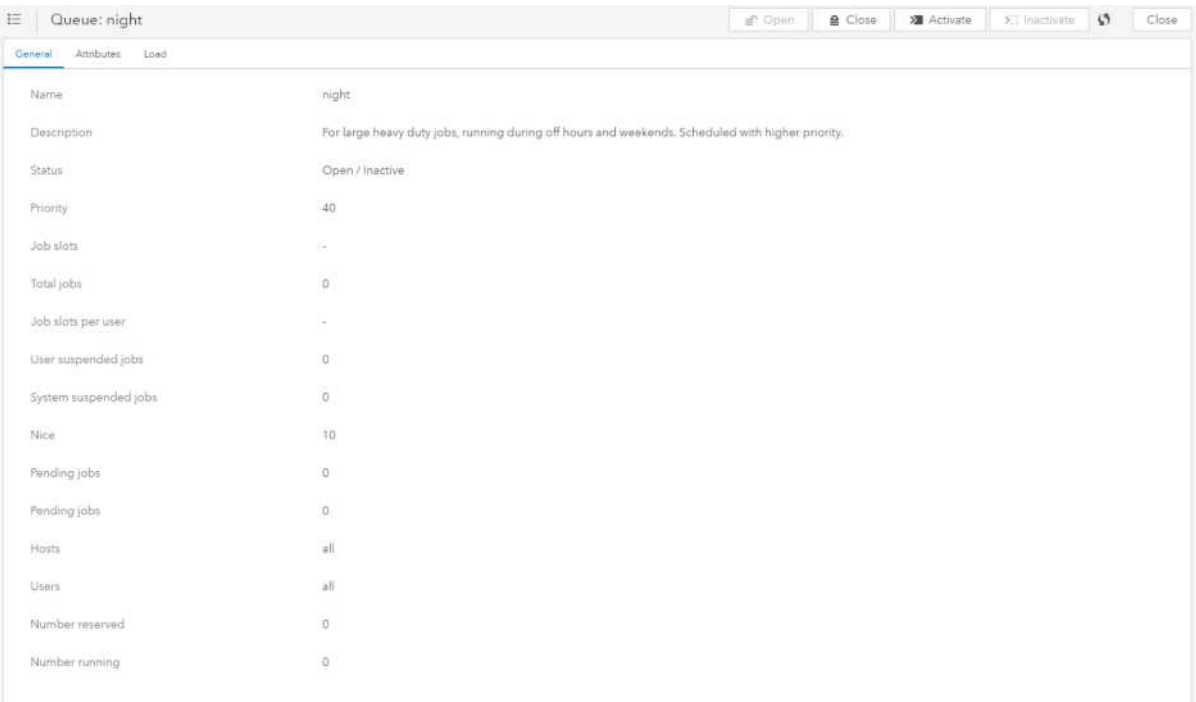
The SAS Grid Manager **Queues** area enables you to manage queues and view detailed information about each queue. Use the **LSF Configuration** area in the SAS Grid Manager for Platform module to define queues.

To view information about a queue, follow these steps:

1 In the left navigation menu, select **Queues**. The **Queues** table appears, and it contains summary information for the queues on the grid.



2 Click an entry in the **Name** column to view the **Queue: *queue_name*** tables for the queue.



3 The **General** tab contains configuration data for the queue. The **Attributes** tab contains any custom attributes that have been defined for the queue. The **Load** tab contains a snapshot of the load on the queue.

To perform actions on queues in the SAS Grid Manager for Platform module, follow these steps:

1 In the left navigation menu, select **Queues**.

2 In the table, select the check box for the queue that you want to perform an action on. Select multiple check boxes to perform an action on more than one queue.

3 Available actions appear at the top of the **Queues** table. Choices for actions are:

Open
Opens a closed queue. The queue can accept new jobs and processes the jobs in the queue.

Close
Closes a queue. A closed queue cannot accept any jobs that are sent to the grid. Closing a queue is useful when you need to make configuration changes.

Activate
Activate an inactivated queue. The queue can accept jobs if it is open, and the jobs in the queue are processed.

Inactivate
Makes a queue inactive. An inactive queue can still accept jobs if it is open, but none of the jobs in the queue can be processed.

Note: Some queues might be defined to be inactive during certain time periods. If you attempt to open a queue during an inactive time period, the action seems to be successful, but the queue remains inactive.

4 Confirm the action when prompted.

Click the refresh icon to manually refresh the table.

Managing Hosts

The SAS Grid Manager for Platform **Hosts** area enables you to view configuration and usage information about hosts, as well as to perform actions on the hosts.

To view information about a host, follow these steps:

1 In the left navigation menu, select **Hosts**. The **Hosts** table appears, and it contains summary information about the hosts on the grid.

2 Click an entry in the **Name** column to view the **Host:*hostname*** page for the host.

3 The **General** tab contains configuration data and load information for the host. The **Resource** tab contains details about the resources available for the host (such as CPU, memory, and disks).

Note: Little or no configuration, resource, or load information is displayed for hosts that are defined as grid client nodes.

To perform an action on a host in SAS Grid Manager for Platform, follow these steps:

1. In the left navigation menu, select **Hosts**.
2. In the table, select the check box for the host that you want to close or open. Select multiple check boxes to perform an action on more than one host.

   Note: Actions are available only for grid server nodes.

3. Available actions appear at the top of the **Hosts** table. Choices for actions are:

   Open
   Enables jobs to be processed on the machine.

   Close
   Prevents jobs from processing on the machine. Closing a host is useful when you want to remove the host from the grid for maintenance.

4. Confirm the action when prompted.

Click the refresh icon to manually refresh the table.

Managing Services

The SAS Grid Manager for Platform module **Services** area enables you to control and view information about high availability applications running on the grid. High availability applications are configured through the **HA configuration** area in the SAS Grid Manager for Platform module, and are defined to have a primary host and a failover host. If the primary host fails, the application automatically starts on the failover host. The area enables you to view the status of high availability applications running on hosts on the grid, to stop applications, and to start applications on either the primary host or the failover host.

To view information about a HA service, follow these steps:

1. In the left navigation menu, select **Services**. The **Services** table appears, and it contains summary information about the defined HA services.
2. Click an entry in the **Name** column to view the **Service:*servicename*** page.
3. The **General** tab displays the service name and version, as well as the status and number of instances. The **Instances** tab contains details (execution host, status, process ID, and start time) about the instances of the service.

To perform an action on an HA service in SAS Grid Manager for Platform, follow these steps:

1. In the left navigation menu, select **Services**.
2. In the table, select the check box for the service that you want to manage. Select multiple check boxes to perform an action on more than one host.

3 Available actions appear at the top of the **Services** table. Choices for actions are:

Start
Start the selected application based on the application definition.

Stop
Stop the selected application.

Restart
Restart the application on the original host if that host is running.

Migrate
Start the selected application on the failover host. Because migrating an application might take a long time, you should migrate only a few applications at one time in order to avoid time-out errors. This option is useful if you are performing maintenance on the primary host.

4 Confirm the action when prompted.

Click the refresh icon to manually refresh the table.

Managing Audit Logs

The SAS Grid Manager for Platform **Audit Logs** area enables you to view information about records in the LSF audit log database.

To view information about a log record, follow these steps:

1 In the left navigation menu, select **Audit Logs**. The **Audit Logs** table appears, and it contains summary information about the log records.

2 Click an entry in the **Name** column to view the **Audit log ID:*logID*** page for the log record.

3 The **General** tab contains general information about the log record, including the time, status code, and action. The **Query parameters** tab lists any specified logging parameters.

By default, audit logs are purged after 30 days. To change the amount of time that audit logs are kept, change the value of the AUDIT_LOG_KEEP parameter in the Platform Web Services database.

Click the refresh icon to manually refresh the table.

Working with LSF Configurations

Select **LSF Configuration** in the left navigation menu to manage sets of LSF parameters that are applied to all resources on the grid. An LSF configuration contains items such as a list of grid machines, definitions of queues that are used on the grid, and grid resource definitions. The initial LSF configuration is contained in the lsf.conf file, and it is created when you install Platform Suite for SAS.

To create or modify an LSF configuration, follow these steps:

1. In the SAS Grid Manager for Platform module, select **LSF configuration** in the left navigation menu. The **LSF Configuration** table appears. The table lists all the LSF configurations that you have created. A configuration that is created from the current settings is indicated by a check mark in the **In use** column. When you use the module for the first time, this table will be blank, because there are no user-created configurations.

2. Click **Load** to use the settings for the currently active configuration in order to create a new configuration. The newly created configuration is marked with a check mark in the **In Use** column.

3. Click on the configurations ID in the **ID** column to edit the configuration settings. The **LSF configuration:*configuration-ID*** window appears. The window is organized into these tabs:

   Settings
   include global settings, cluster settings, batch settings, EGO settings

   Queues
   define and edit the queues for the cluster. The information for this tile is read from and written to the lsb.queues file.

   Cluster Hosts
   change the hosts in the cluster and in the resource assignments for each host. The information for this tile is read from and written to the lsf.cluster.*cluster_name* file. Expand the entry for a host in this tab to change the settings for the host.

   Batch Hosts
   configure the host and job processing for the cluster. The information for this tile is read from and written to the Host section of the lsb.hosts file.

   Host Groups
   configure the host groups in the cluster. The information for this tile is read from and written to the HostGroup section of the lsb.hosts file.

   Users
   define the LSF users in the cluster. The information for this tile is read from and written to the User section of the lsb.users file.

   User groups
   define the LSF user groups in the cluster. The information for this tile is read from and written to the UserGroup section of the lsb.users file.

   Resources
   define and edit common resource definitions that are shared by all defined clusters. The information for this tile is read from and written to the lsf.shared file.

   Resource limits
   define limits for how many resources must be available in order for different classes of jobs to start and which resource consumers the limits apply to. The information for this tile is read from and written to the Limits section of the lsb.resources file.

   Cluster administrators
   define the users who can administer the cluster resources. The primary administrator is specified during installation and cannot be changed. The information for this tile is read from and written to the lsf.cluster.*cluster_name* file.

4 Click the name of an entry in the page to edit a definition. Click the **Add** icon + in the table to add a new entry.

5 When you finish changing the configuration values, click to save your changes and return to the configuration list. You can also click to return to the table. You are prompted to save any changes.

6 Verify that all hosts in your cluster are available. If they are not available, make these changes to avoid configuration problems:

- In the Postgres database table `public.pws_action_status`, change the value of the `action_message` column to 4096.
- If your LSF host is not on a fast machine or if your grid has many nodes, you can also change the `PWS_LIM_RECONFIG_TIME` column in the Postgres table `public.pws_configuration_table` from the default value of 10 to a value of 30.

7 Select the check box beside the configuration you just modified and click **Apply**. The SAS Grid Manager for Platform module writes the configuration settings to the configuration files. LSF then applies the settings in the files to the grid cluster.

8 To rename a configuration. select the check box beside the entry in the table and select **Rename**.

9 To delete a configuration, select the check box beside the entry in the table and select **Delete**. Click the header for the check box column to select all entries.

Working with High Availability Configurations

Creating an HA Configuration

Select **HA configuration** in the left navigation area to manage configuration and operation of high availability applications on the grid. An HA configuration contains information to enable applications to automatically start on a failover server if the primary server goes down.

To create or modify a HA configuration, follow these steps:

1 In the SAS Grid Manager for Platform module, select **HA configuration** in the left navigation area. The **HA Configurations** table appears. The table lists all the HA configurations that you have created. A configuration that is created from the current settings is indicated by a check mark in the **In use** column. When you use the module for the first time, this table will be blank, because there are no user-created configurations.

2 Click **Load** to use the settings for the currently active configuration in order to create a new configuration. The newly created configuration is marked with a check mark in the **In Use** column.

3 Click anywhere in the configuration's row to edit the configuration settings. The **High-availability configuration: *configuration_name*** page appears. Each entry on the page represents a configuration for an HA service.

4 Click the Add icon + to create a new configuration for an HA service.

5 In the Add service window, specify the name of the service and the mode of the service. These are the choices for the **Mode** field:

Active-passive
The service is started on the primary host and starts on the failover host only if the primary host fails. If you select this value, you must also specify a **Primary host** and a **Failover host**.

Active-active
The service is started on all selected hosts. If you select this value, you must select the defined hosts on which the service is started.

6 Use the **Start type** field to specify whether the service starts automatically or manually on the hosts. Provide the script that is used to start the service in the **Start command** field, and provide the user ID under which the service runs in the **Execution user** field. Click **OK** to define the service.

7 After you define the service, use the fields on the **High-availability configuration: *configuration_name*** page to define additional parameters, such as the number of restarts for the service script, the failover interval, and any dependencies (applications that either depend on the HA service or applications that the HA service depends on). You can also define affinity requirements for the service.

Affinity requirements specify where a selected application runs in relation to other applications in the HA configuration. The XML definition of an affinity application must exist in the EGO system. This option is available only if you are using LSF 10. Here are the options:

- The selected application must run on the same server as the other applications. This is known as hard affinity.
- The selected application must run on a different server than the other applications. This is known as hard anti-affinity.
- The selected application should run on the same server as the other applications, if possible. This is known as soft affinity.
- The selected application should run on a different server than the other applications, if possible. This is known as soft anti-affinity.

8 When you finish changing the values for the service, save the configuration to return to the service list. Continue adding applications as needed. When you have finished adding all applications, click ⁝☰ to return to the **HA Configurations** table.

9 Select the check box beside the configuration that you just modified and click **Apply**. The SAS Grid Manager module writes the configuration settings to the configuration files. LSF then applies the settings in the files to the grid cluster.

Specifying HA Application Dependencies

When you define an HA application configuration, you can specify a dependent application. A dependent application has either of these requirements:

- is started or continues to run based on the state of the HA application
- must be started before the HA application starts

If you are using LSF10, follow these steps to define a dependency for an HA application.

1. On the Application window for a new or existing HA configuration, click **Dependencies**.
2. The **Dependencies** table lists the dependencies. If no dependencies are defined, select **On start** or **Conditional** to choose which type of dependency to use for the configuration. If dependencies are already defined, the new dependency must be of the same type. All dependencies must be of the same type for an application's configuration.

   An **On start** dependency does not start until the HA application starts.

   A **Conditional** dependency specifies that the dependent application starts or keeps running according on the state of the HA application.
3. To specify a new **On start** dependency, select the **Enabled** option for the dependent application.
4. To specify a new **Conditional** dependency, select the **Enabled** option for the dependent application. You must specify a value for both **Keep State** and **Satisfy State** fields.

   Satisfy states
   The HA application must be in one of the states that you select in the **Satisfy states** column in order for the dependent application to start. For example, if you select **Active** in the **Satisfy states** column, the HA application must be running in order for the dependent application to start.

   Keep states
   The HA application must be in one of the states that you select in the **Keep states** column in order for the dependent application to keep running. For example, if you select **Frozen** and **Inactive** in the **Keep states** column, the dependent application continues to run as long as the HA application is either frozen or inactive.

   Auto start is turned on
   Starting the dependent application causes the HA application to start.

   The following example shows a dependency on **sastest**, **Active** for **Satisfy states**, and **Active** and **Tentative** for **Keep states**.

Name	Enabled	Satisfy states	Keep states	Auto start
sastest		☐ Starting ☐ Stopping ☑ Active ☐ Inactive ☐ Error ☐ Frozen ☐ Tentative	☐ Starting ☐ Stopping ☑ Active ☐ Inactive ☐ Error ☐ Frozen ☑ Tentative	

5 If you are defining an **onStart** dependency, use the **Application Name** field to select the dependent application. You can choose applications for which you have already configured an HA application definition.

6 Select to save the dependency.

If you are using LSF 9, only **On start** dependencies are supported. Select the applications that should not start until the HA application starts. The applications are automatically assigned as **On start** dependencies.

Updating the LSF License

When you update your SAS license, you must also follow a separate procedure to update your LSF license. Using the SAS Grid Manager for Platform module for SAS Environment Manager, select **Utilities** in the left navigation menu to update your LSF license. You can use this function to update the LSF license with either a SAS license or an IBM entitlement file.

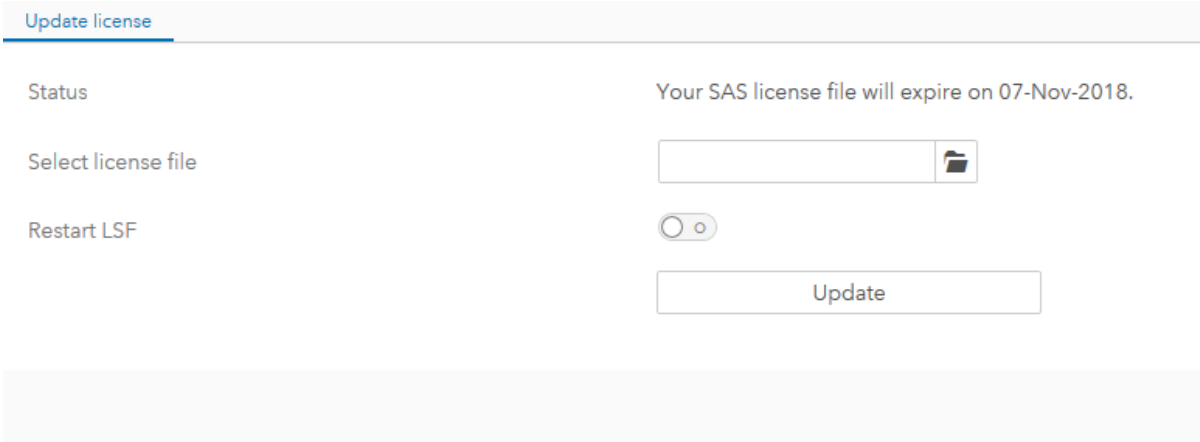
Follow these steps to update your LSF license.

1 Ensure that these requirements are met:

Windows
: The user account that is used to start the Platform Web Services tcServer instance must have Write permission for the LSF license file in the LSF cluster.

UNIX and Linux
: You must configure passwordless SSH between the Platform Web Services host and the LSF master host for the user account that is used to start the Platform Web Services tcServer instance. You must configure passwordless SSH to enable the Platform Web Services user to SSH to use the LSF master host as the primary LSF administrator without a password prompt. An example is `ssh primary_admin@lsf_master`.

2 In the SAS Grid Manager for Platform module, select **Utilities** in the left navigation menu. The Update License page displays the current status of your license.



3 Specify or navigate to the location of your LSF license file in the **Select license file** field. Select the file in the form `SAS94_xxx.txt`, where *xxx* specifies the grid platform. The contents of this file are then stored in the LSF configuration directory as the file `license.dat`.

4 If you are using UNIX or Linux, select **Restart LSF** to specify that the LSF services should restart after you update the license file. Select this option if you want the update to take effect immediately. If you are using Windows, an administrator must restart the LSF services. A restart is typically accomplished by using the Windows command line.

5 Click **Update**.

Using the SAS Grid Manager for Platform Module as the LSF Administrator

These SAS Grid Manager for Platform module functions are available only for the LSF Administrator user:

- **Services** area
- **Audit Log** area
- **LSF Configuration Manager** area
- **HA Configuration Manager** area
- Ability to view job information for all users

Users in the SAS Grid Manager for Platform module are mapped to users that are created in SAS metadata. In order for the LSF Administrator user identity to be available as a valid user in the SAS Grid Manager for Platform module, you must perform one of the following actions:

- If the LSF Administrator identity is defined in metadata but not in SAS Environment Manager, then add the identity to the group SAS Environment Manager Super Users in SAS metadata and synchronize the users in SAS Environment Manager (select **Synchronize Users** under the **Manage** tab).

- If the LSF Administrator identity is defined in metadata and in SAS Environment Manager, select **Manage** ⇨ **List Users**, select the user identity, then click **Add to List** to add the identity to the **Super User Role**.
- If identities have been defined in the SAS Environment Manager Super Users group, then you can specify one of those identities as an LSF Administrator by adding the identity to the LSF configuration file. You do not have to synchronize the SAS metadata and SAS Environment Manager user information if you use this approach.

Adding the LSF Administrator user to the SAS Environment Manager Super Users group enables the user to sign on to the SAS Grid Manager for Platform module using the LSF Administrator's credentials and assigns the user to the Super User role in SAS Environment Manager. See *SAS Environment Manager: User's Guide* for information about user management in SAS Environment Manager.

Using Platform RTM for SAS

Note: This topic applies only to older installations that used Platform RTM for SAS. In SAS 9.4M3, a SAS Grid Manager Module for SAS Environment Manager was added. The Grid Manager Module for SAS Environment Manager serves the function previously provided by Platform RTM for SAS.

Platform RTM for SAS is a web-based tool that lets you graphically view the status of devices and services within your SAS Grid Manager for Platform environment as well as manage the policies and configuration of the grid. It is a visual tool to quickly track and diagnose issues before they affect service levels.

IMPORTANT Platform RTM is not part of Platform Suite for SAS. The version of RTM that was available as a download for use with SAS Grid Manager is in Limited Support. See SAS Technical Support Services and Policies for more information about Limited Support. Contact SAS Technical Support for information about RTM.

Platform RTM for SAS provides these features:

- drill-down capabilities to view details of hosts, jobs, queues, and user activities
- instant alerts on job performance and grid efficiency to enable administrators to optimize usage and workloads
- customizable graphs to visually analyze resource usage, workload trends, and job behavior
- interfaces to allow administrators to update the policies and rules on the grid configuration as well as set up high availability for any critical grid services executing on the grid

Platform RTM for SAS helps system administrators improve decision-making, reduce costs, and increase service levels for SAS Grid Manager for Platform deployments. Refer to the documentation included with the Platform RTM for SAS installation package for instructions.

Platform RTM for SAS is supported on the following operating systems:

- Red Hat Linux 32/64 bit
- Windows Server 2008 R2 64 bit
- Windows 7 32/64 bit

If you have a UNIX grid, you must install the Linux version of Platform RTM for SAS on a Linux machine or a Linux virtual machine (VM). If you have a Windows grid, you must install the Windows version of Platform RTM for SAS on a Windows 7 or Windows Server 2008R2 machine or virtual machine (VM).

Note: Although you can use either the Grid Manager Module for SAS Environment Manager or Platform RTM for SAS to modify your LSF configuration, you cannot alternate between the two. There is no communication or synchronization between the Grid Manager Module and Platform RTM for SAS. For example, if you specify an LSF configuration as active in the Grid Manager Module and then use Platform RTM for SAS, Platform RTM for SAS does not know that the LSF configuration has been modified and consequently works with the wrong parameters and values.

Using the Grid Manager Plug-in for SAS Management Console

Overview

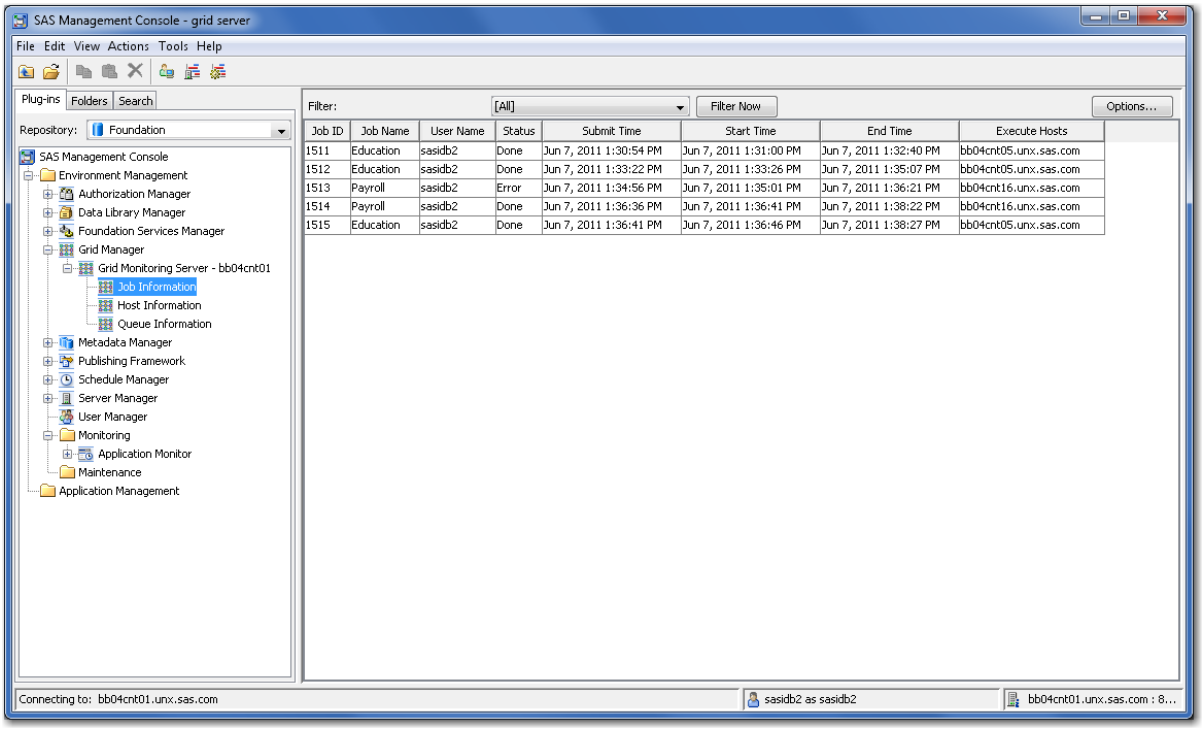
Note: Starting in SAS 9.4M7, Grid Management Services (GMS) is removed from Platform Suite for SAS. GMS version 8.01, which was delivered with SAS 9.4M6, is not compatible with Platform Suite for SAS 11.1 (including Process Manager 10.2 and LSF 10.16), which was also delivered with SAS 9.4M6. Therefore, GMS and the Grid Manager plug-in for SAS Management Console are in Limited Support. See SAS Technical Support Services and Policies for more information about Limited Support.

The Grid Manager plug-in for SAS Management Console enables you to monitor SAS execution in a SAS Grid Manager for Platform environment (the plug-in does not work with SAS Grid Manager for Hadoop). This plug-in enables you to manage workloads on the grid by providing dynamic information about the following:

- jobs that are running on the grid
- nodes that are configured on the grid
- job queues that are configured on the grid

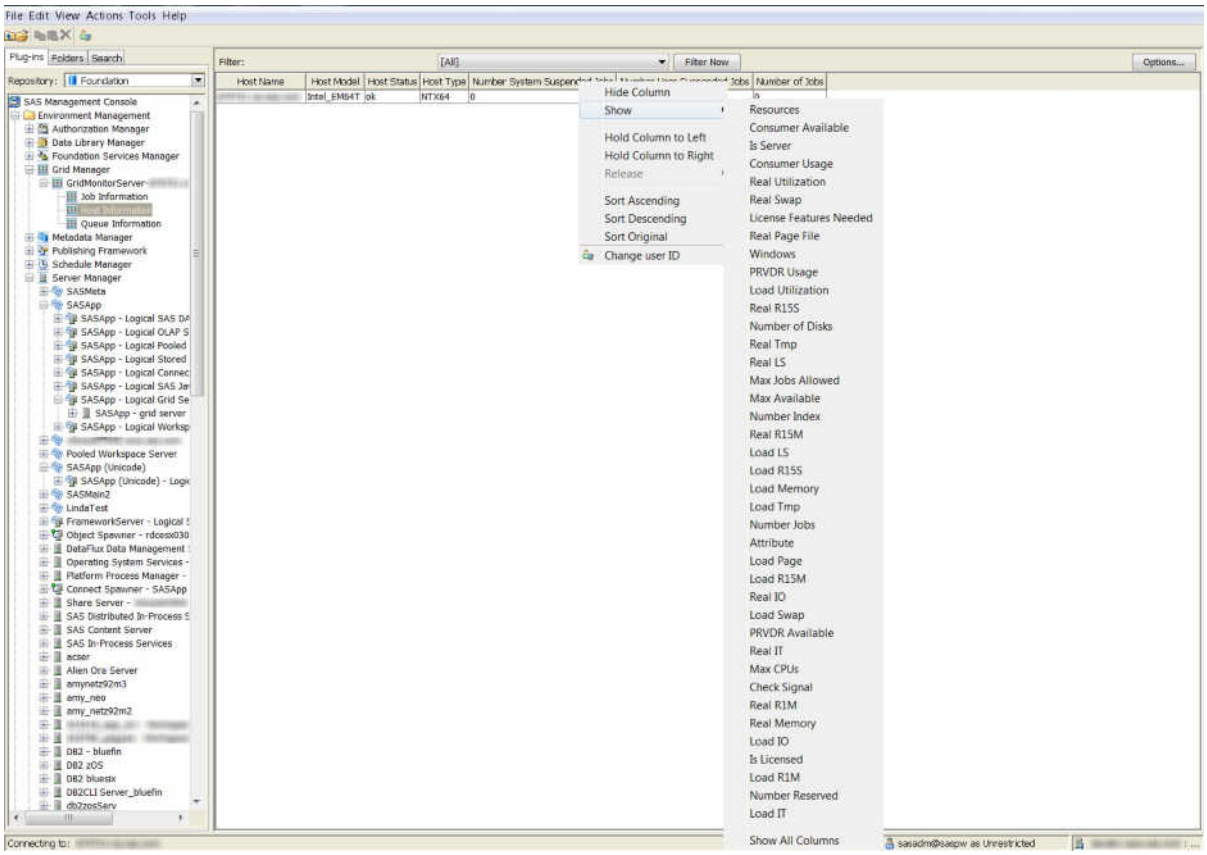
Information is displayed in tabular or chart format. Here is an example of a job view:

Figure 8.1 *Job View in the Grid Manager Plug-in to SAS Management Console*



Using the SAS Grid Manager, you can customize the view by selecting the columns of data to display and the order in which they should appear. In addition, you can filter, sort, and refresh the display of jobs.

Figure 8.2 Subsetting Data in the Grid Manager Plug-in



Each grid that you define must have one grid monitoring server configured and running on a machine on the grid.

Note: Starting with SAS 9.4M5, job names that use a double-byte character set (DBCS) are not converted to Escaped Unicode and are unreadable in the Grid Manager plug-in for SAS Management Console. They are readable in the Grid Manager module for SAS Environment Manager. You must set the `SAS_GRID_ESC_UNICODE` environment variable to 1 in order to make the names readable in SAS Management Console.

Maintaining the Grid

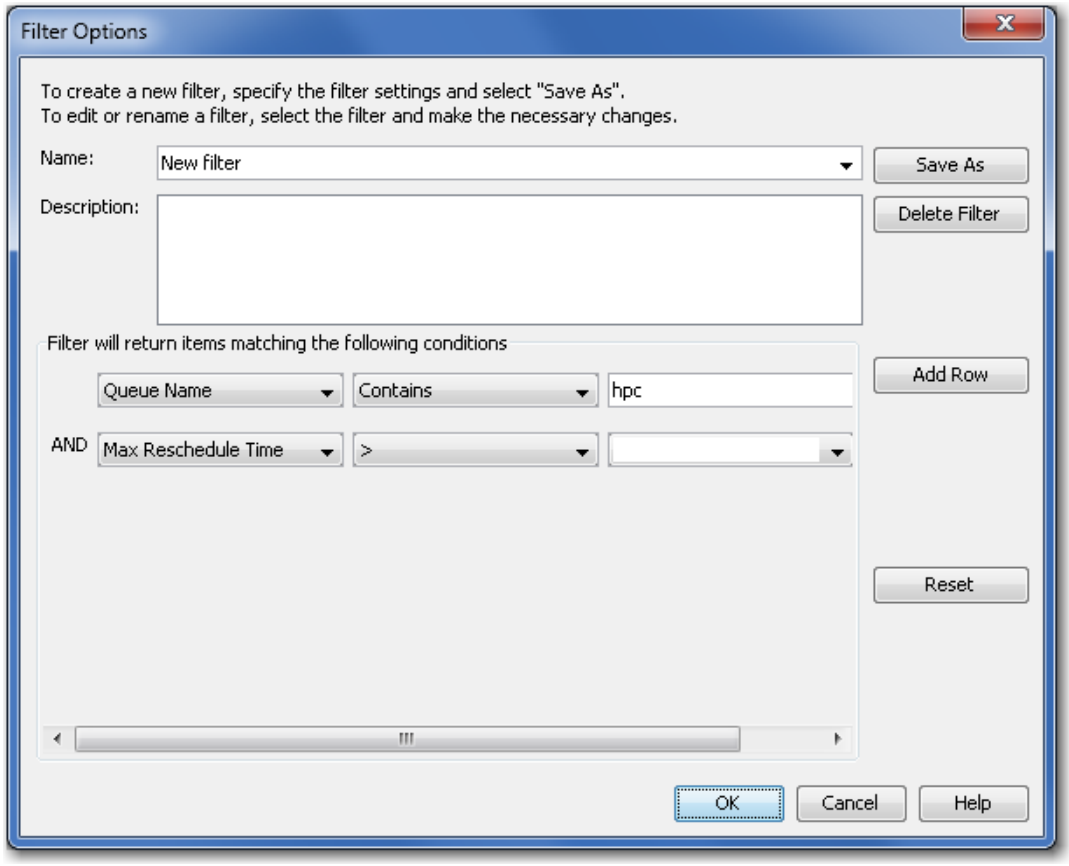
Viewing Grid Information

When you expand the **Grid Manager** node in the navigation tree, all of the grid monitoring servers that you have defined are listed under the name of the plug-in. Each one represents a unique grid. To view information about a specific grid, expand the server's node in the navigation tree. The information for a grid is grouped into three categories in the navigation tree:

- Job Information
- Host Information
- Queue Information

Select a category to display a table that contains information for the category. You can also display a graph of the job information. Click the column headings to select the information that is displayed in the table. Click **Options** to start the Filter wizard, which you can use to select which jobs to display.

Figure 8.3 *Filter Options Dialog Box*



After you have defined filters, you can select a filter and click **Filter Now** to filter the displayed information. You can also manage jobs, hosts, and queues from the tables.

Right-click the **Grid Monitoring Server** node in the navigation tree and select **Options** to specify that the information from the grid is automatically refreshed and how often it is refreshed.

Managing Jobs

Use the Grid Manager to terminate or suspend running jobs and terminate or resume suspended jobs.

To terminate a job, follow these steps:

1. In the selection tree, select the **Job Information** node.
2. In the table, locate the job that you want to cancel.
3. Right-click any column in the row for the job and select **Terminate Task** from the pop-up menu.

If you log on to SAS Management Console using a user ID that is defined as an LSF Administrator ID, you can terminate any jobs that have been submitted to the grid. Users can terminate only their own jobs. The LSF Administrator can terminate any job. If you are terminating a job on Windows, be sure to match the domain name exactly (including case).

To suspend a job (pause the job's execution), follow these steps:

1. In the selection tree, select the **Job Information** node.
2. In the table, locate the job that you want to suspend.
3. Right-click any column in the row for the job and select **Suspend Job** from the pop-up menu.

To resume processing of a suspended job, follow these steps:

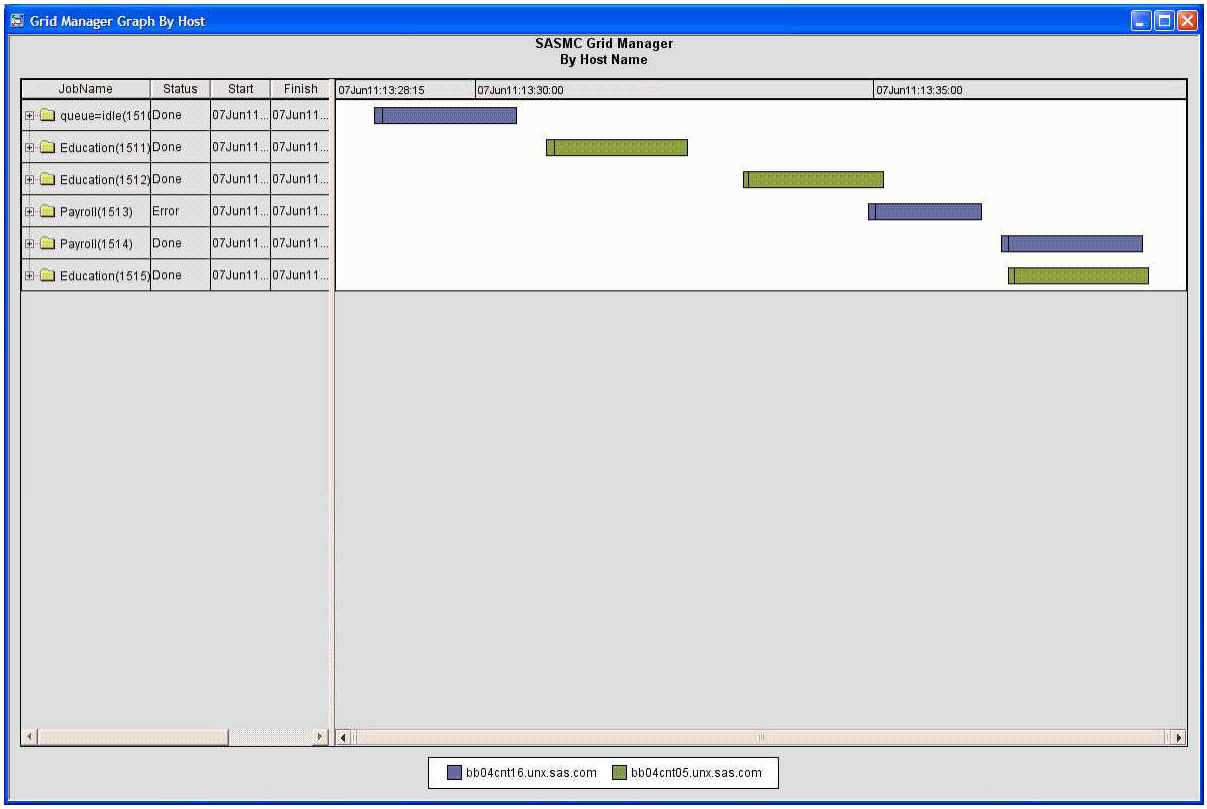
1. In the selection tree, select the **Job Information** node.
2. In the table, locate the job that you want to resume.
3. Right-click any column in the row for the job and select **Resume Job** from the pop-up menu.

Displaying Job Graphs

You can use the Grid Manager to display GANTT charts for jobs running on the grid. To display a chart, follow these steps:

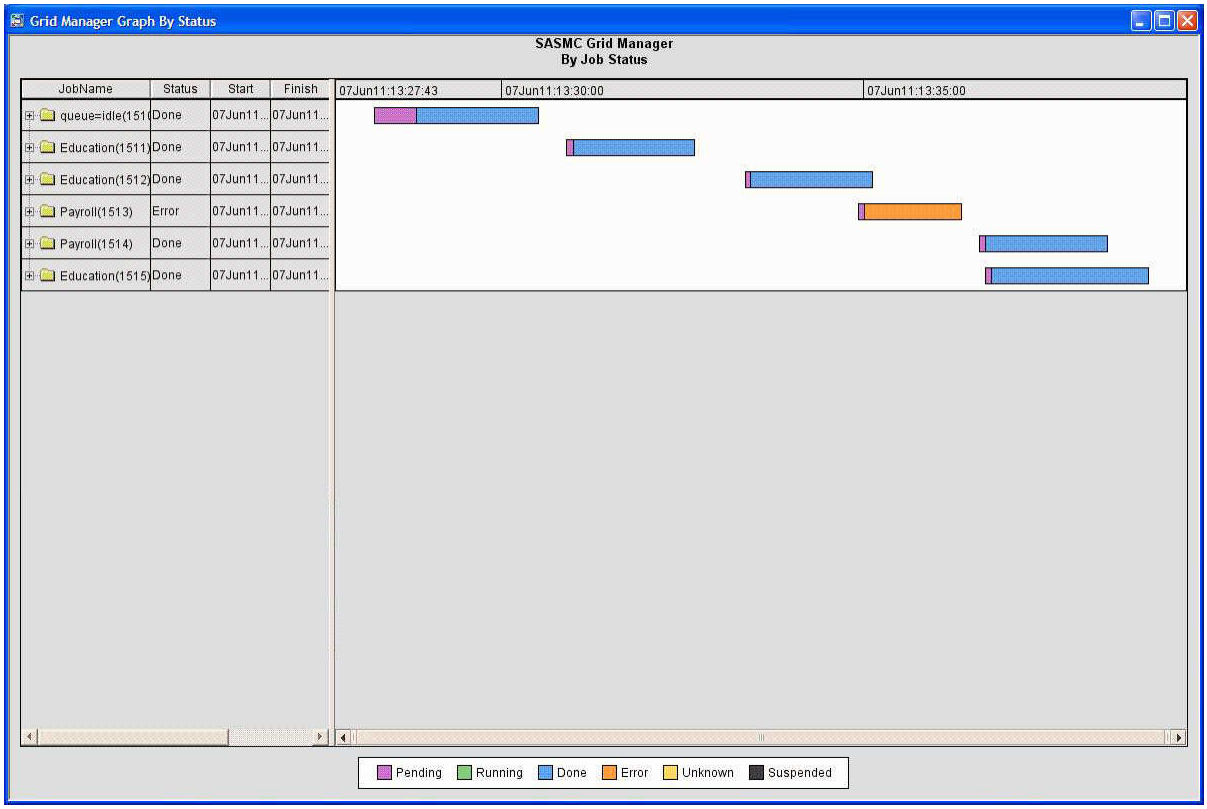
1. In the selection tree, select the **Job Information** node.
2. Right-click and select either **Create Graph by Host** or **Create Graph by Status** from the **Actions** menu, the pop-up menu, or the toolbar.
3. Select **Create Graph by Host** to display a Gantt chart that shows the amount of time taken to process each job and identifies the machine on which the job ran.

Figure 8.4 *Display of Grid Jobs by Host*



4 Select **Create Graph by Status** to display a Gantt chart that illustrates the amount of time that each submitted job spent in each job status (such as pending or running).

Figure 8.5 Display of Grid Jobs by Status



Closing and Reopening Hosts

You can use the Grid Manager to close or reopen hosts on the grid. A closed host cannot process any jobs that are sent to the grid. Closing a host is useful when you want to remove the host from the grid for maintenance. You can also close the grid control server to prevent it from receiving work.

Note: The status of a host does not change right away after it has been opened or closed. By default, the host status is polled every 60 seconds by the Grid Management Service. The polling time interval is specified by the GA_HOST_POLL_TIME property in the ga.conf file, which is located in the ***LSF_install_dir*/gms/conf** directory.

To close a host, follow these steps:

1. In the navigation area, open the node for the grid containing the host.
2. Select the **Host Information** node.

   The display area contains a table of the hosts on the grid.
3. In the table, right-click the host that you want to close and select **Close** from the pop-up menu.

The host now cannot accept jobs that are sent to the grid.

To open a host that has been closed, follow these steps:

1 In the navigation area, open the node for the grid containing the host.

2 Select the **Host Information** node. The display area contains a table of the hosts on the grid.

3 In the table, right-click the host that you want to open and select **Open** from the pop-up menu. The host can now accept jobs that are sent to the grid.

Managing Queues

You can use the Grid Manager to close, open, activate, and inactivate queues. A closed queue cannot accept any jobs that are sent to the grid. An inactive queue can still accept jobs, but none of the jobs in the queue can be processed. Closing a queue is useful when you need to make configuration changes to the queue.

To close a queue, follow these steps:

1 In the navigation area, open the node for the grid containing the queue.

2 Select the **Queue Information** node.

The display area contains a table of the queues on the grid.

3 In the table, right-click the queue that you want to close and select **Close** from the pop-up menu.

The queue is now prevented from accepting jobs that are sent to the grid.

To open a closed queue, follow these steps:

1 In the navigation area, open the node for the grid containing the queue.

2 Select the **Queue Information** node.

The display area contains a table of the queues on the grid.

3 In the table, right-click the queue that you want to open and select **Open** from the pop-up menu.

The queue can now accept jobs that are sent to the grid.

To inactivate a queue, follow these steps:

1 In the navigation area, open the node for the grid containing the queue.

2 Select the **Queue Information** node.

The display area contains a table of the queues on the grid.

3 In the table, right-click the active queue that you want to make inactive and select **Inactivate** from the pop-up menu.

To activate a queue, follow these steps:

1 In the navigation area, open the node for the grid containing the queue.

2 Select the **Queue Information** node.

The display area contains a table of the queues on the grid.

3 In the table, right-click the inactive queue that you want to make active and select **Activate** from the pop-up menu.

9

High Availability in a SAS Grid Manager for Platform Grid

High Availability and SAS Grid Manager for Platform

Your organization might have services and long-running SAS programs that are critical to your operations. The services must be available at all times, even if the servers that are running them become unavailable. The SAS programs must complete in a timely manner, even if something happens to cause them to fail. For a SAS program that takes a long time to run, this means that the program cannot be required to restart from the beginning if it ends prematurely.

SAS Grid Manager for Platform provides high availability through these capabilities:

- Multi-machine architecture. Because how a SAS grid is configured and operates, there is no single point of failure. Because jobs are processed on the available grid nodes, if a node becomes unavailable other nodes can take over the workload.
- Platform Suite for SAS. The default configuration of Platform Suite for SAS provides high availability for the grid operation. The LSF master daemon runs on a specified grid node (usually the grid control server), and a failover node is also identified. If the master daemon node fails, the failover node automatically takes over and broadcasts to the rest of the grid. The grid recognizes the new master daemon node and continues operation without interruption. Platform PM and GMS must be treated as critical services and configured for failover along with all other critical services.

Note: Starting in SAS 9.4M6, Grid Management Services (GMS) is not updated to work with the latest release of LSF. Therefore, GMS is in Limited Support.

- Critical service failover. There are certain services and processes that are critical to the operation of SAS applications on the grid and that must always be available (for example, the SAS Metadata Server). After providing a failover host for the service, you can use Platform Computing's Enterprise Grid Orchestrator (EGO) to monitor the service, restart the service if it stops, and start the service on the failover host when needed. Once the service has started on the failover host, you can use either hardware (a load balancer) or software (EGO) to automatically direct clients to the failover host. EGO is part of Platform Suite for SAS that is included with the SAS Grid Manager for Platform and installed as part of the LSF installation process.
- Automatic SAS program failover. If a long-running SAS job fails before completion, rerunning it from the beginning can cause a loss of productivity. You can use the SAS Grid Manager Client Utility to specify that the job is restartable. This means that a failed job restarts from the last successful procedure, DATA step, or labeled section. This capability uses the SAS checkpoint and restart functions to enable failed jobs to complete without causing delays. You can also use attributes on the queue definitions on the grid to automatically restart and requeue any job that ends with a specified return code or that terminates due to host failure. Using these options together ensures that critical SAS programs always run successfully and in a timely manner, even if they encounter problems.

All of these strategies are independent of one another, so you can implement the ones that provide the greatest benefit to your organization.

Setting Up High Availability for Critical Applications

On a grid, there are certain services that always need to be available and accessible to clients. These services are vital to the applications running on the grid and its ability to process SAS jobs. Examples include:

- SAS Metadata Server
- SAS object spawner
- Platform Process Manager
- Platform Grid Management Service
- web application tier components

Configuring a grid that provides high availability for these services requires these components:

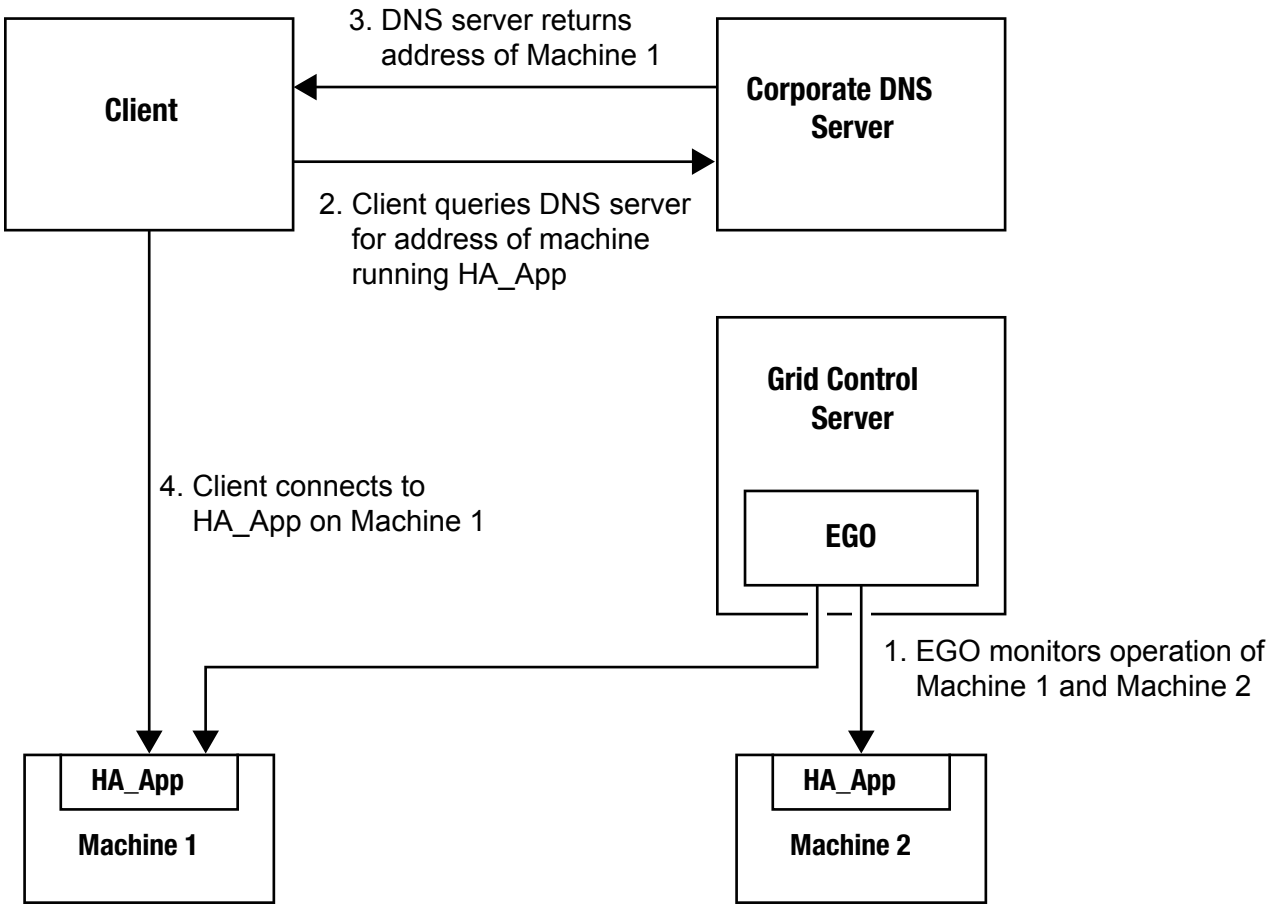
- providing failover hosts for machines that run critical applications. Using multiple machines for critical functions eliminates a single point of failure for the grid.

- providing a way to monitor the high-availability applications on the grid and to automatically restart a failed application on the same host or on a failover host if needed.
- providing a method to let the client know to connect to the failover host instead of the regular host. This can be done through software (DNS resolution) or hardware (the hardware load balancer), but only one is used.

In normal operations, the following sequence takes place:

1. The client determines that it needs to access a service on a machine on the grid.
2. The client sends a query to the corporate DNS server. The DNS server looks up the address for the machine and returns that information to the client.
3. The client uses the address to connect to the machine and use the application.

Figure 9.1 *Normal Grid Operations*



To provide business continuity for the application, a failover host must be provided for the critical services running in the grid environment. This provides an alternative location for running the critical services and ensures that it remains available to the applications on the grid. In addition, both the main and failover machines must have access to a shared file server. This ensures that the application has access to the data required for operation, regardless of which machine is running the service.

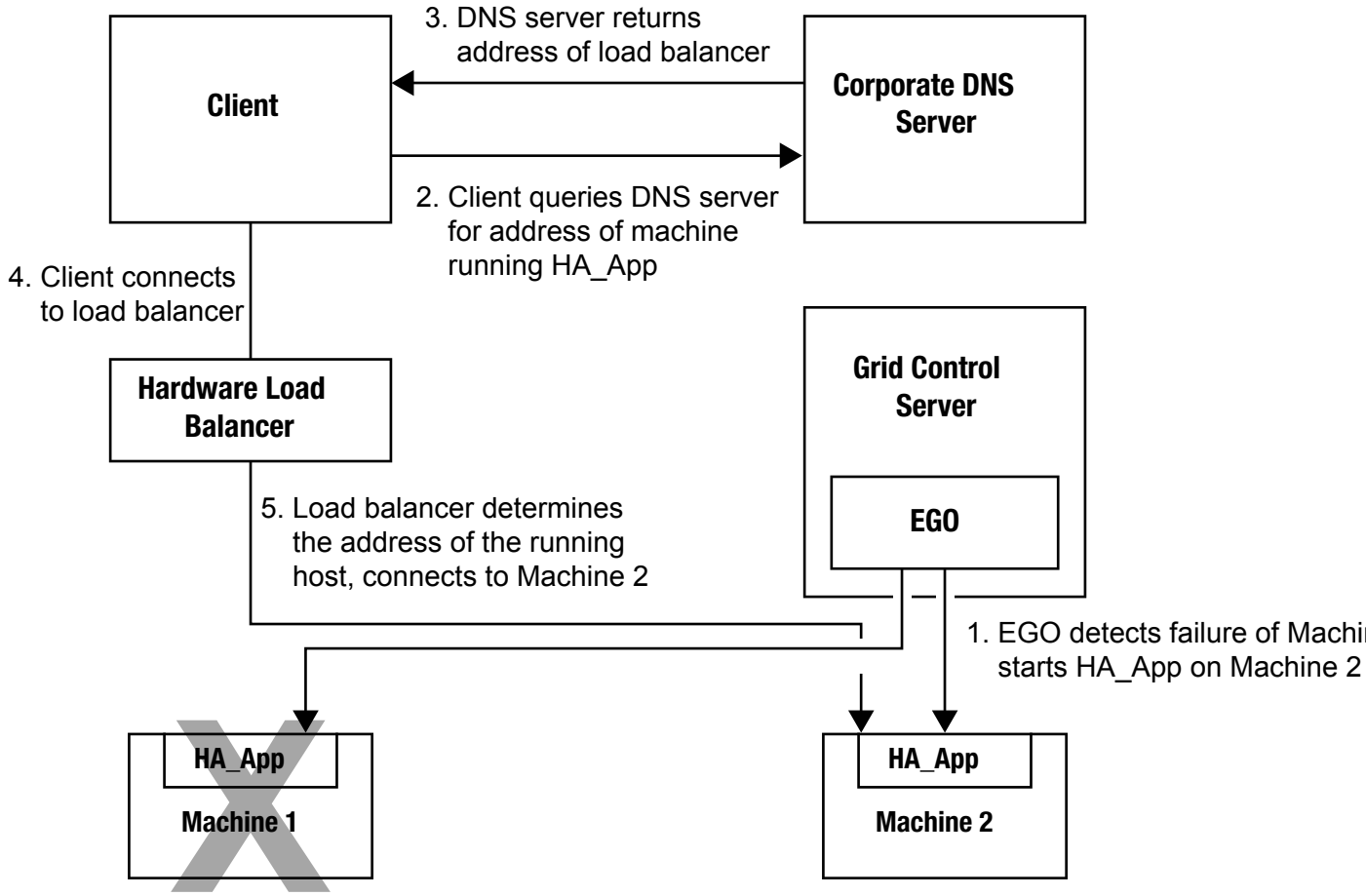
To provide business continuity for the application, the failover capability must also be automatic. EGO is configured to monitor any number of critical services running on the grid. If it detects that the application has failed or that the machine running it

has gone down, it is configured to start the application on the failover server automatically, which enables applications to continue running on the grid.

However, once the application has started on the failover server, the client must have a way to know which server is running the application. There are two methods for accomplishing this:

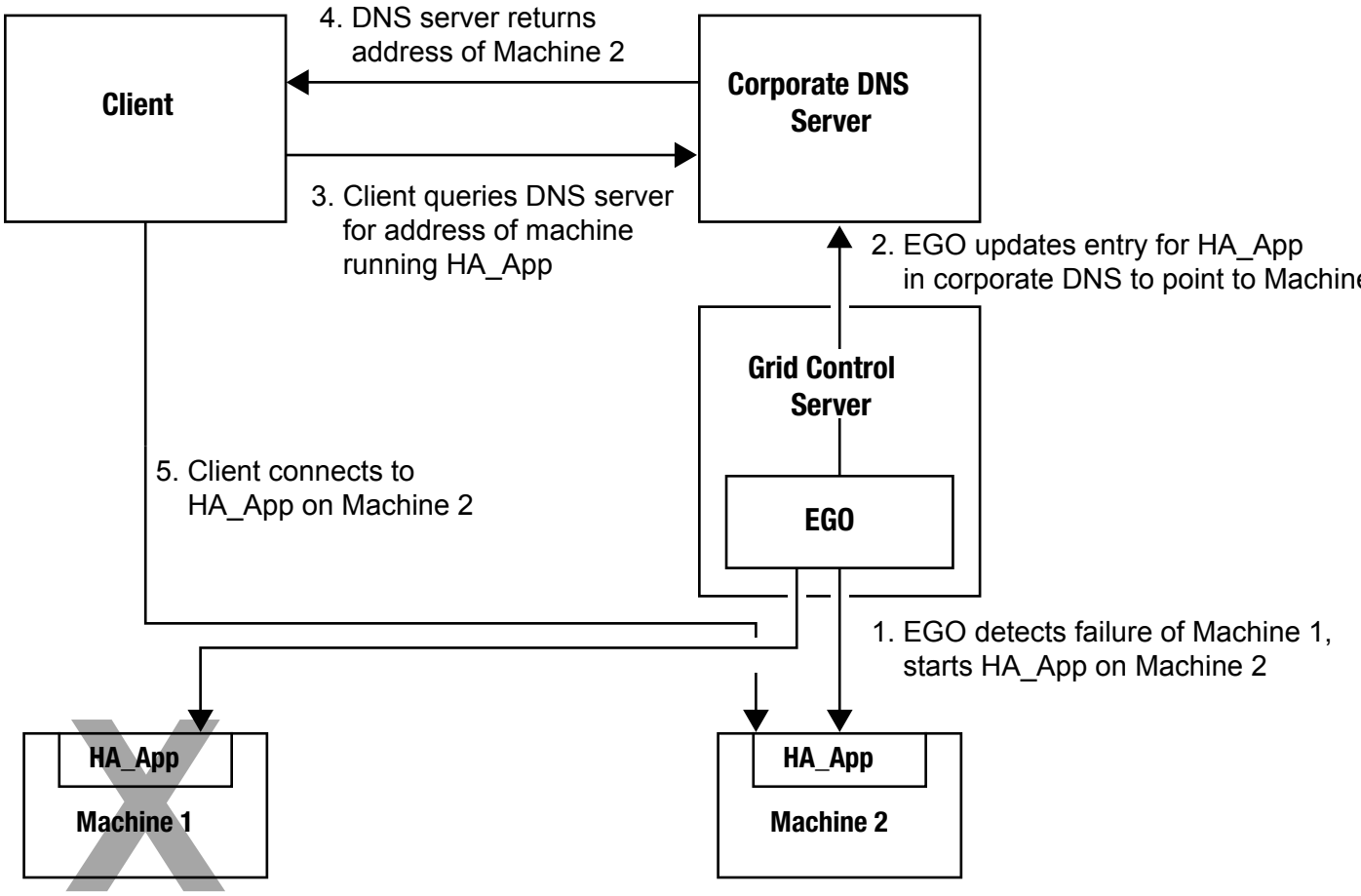
- Using a hardware load balancer. The load balancer serves as an intermediary between the client and the services running on the grid, which decouples the grid operation from the physical structure of the grid. When the client wants to connect to the service, it connects to the load balancer, which then directs the request to the machine that is running the service. The load balancer knows the addresses of both the main and failover machines, so it passes the request on to whichever of the machines is running in the servers. During normal operation, the request goes to the main machine. When failover occurs, EGO starts services on the failover host, and the load balancer forwards connections to it (because it is not the host running the services).

Figure 9.2 *Grid Failover with a Load Balancer*



- DNS resolution. Once EGO starts the application on the failover server, it sends the address of the failover machine to the corporate DNS server. The entry for the application is updated in the server, so the next time a client requests a connection to the application, the DNS server returns the address of the failover machine.

Figure 9.3 Grid Failover with EGO



If you do not want EGO to directly update the corporate DNS, you can configure the DNS server to always point to EGO to provide the IP address for the machine. When EGO starts the application on the failover machine, it then points to the new machine.

The choice of whether to use a load balancer or a DNS solution depends on your organization's policies. Using DNS resolution prevents you from having to purchase an addition piece of hardware (the load balancer). However, your organization's policies might prohibit either the corporate DNS from being changed by an outside DNS (EGO) or DNS requests to be forwarded to an outside DNS. If this is the case, the hardware load balancer provides a high-availability solution.

10

Troubleshooting SAS Grid Manager for Platform

Overview of the Troubleshooting Process

These topics provide the framework for a systematic, top-down approach to analyzing problems with a SAS Grid Manager for Platform grid environment. By starting at the highest level (the network) and working downward to the job execution, many common problems can be eliminated.

For the troubleshooting information not contained here, contact SAS Technical Support.

Verifying the Network Setup

Overview

The first step in troubleshooting problems with a SAS Grid Manager for Platform grid is to verify that all computers on the grid can communicate with one another through the ports that are used by Platform Suite for SAS.

Host Addresses

Check the /etc/hosts file on each grid node to ensure that the machine name is not mapped to the 127.0.0.1 address. This mapping causes the sign-on connection to the grid node to fail or to hang. This happens because the SAS session that is being invoked on the grid node cannot determine the correct IP address of the machine on which it is running. A correct IP address must be returned to the client session in order to complete the connection. For example, delete the name "myserver" if the following line is present in the /etc/hosts file:

```
127.0.0.1 myserver localhost.localdomain localhost
```

Host Connectivity

You must verify that the network has been set up properly and that each machine knows the network address of all the other machines on the grid. Follow these steps to test the network setup:

1. Run the **`hostname`** command on every machine on the grid (including grid nodes, grid control servers, and SAS Foundation grid clients).
2. Run the **`ping`** command on all grid node machines and the grid control machine against every other machine on the grid (including grid client machines). When you ping a grid client machine, use the host name without the domain suffix.
3. Run the **`ping`** command on each grid client machine against every other machine on the grid (including itself). When a grid client machine pings itself using the value from the

   `hostname`

command, verify that the returned IP address is the same IP address that is returned when the grid nodes ping the client. However, this might not occur on machines with multiple network adapters.

If the network tests indicate a problem, you must either correct the DNS server or add entries to each machine's hosts file. Contact your network administrator for the best way to fix the problem.

Platform LSF assumes that each host on the grid has a single name, that it can resolve the IP address from the name, and that it can resolve the official name from the IP address. If any of these conditions are not met, LSF needs its own `hosts` file, which is located in its configuration directory (LSF_ENVDIR/conf/hosts).

Host Ports

You must verify that the ports that SAS and LSF use for communication are accessible from other machines. The ports might not be accessible if a firewall is running on one or more machines. If firewalls are running, you must open ports so that communication works between the LSF daemons and the instances of SAS.

To determine whether a port is open on a specific host, issue the following command:

```
nmap -p- remote_host | grep port
```

Where:

- *remote_host* is the IP address of the remote host.
- *port* is the port to check on the remote host.

The default ports used in a grid include the following:

- LSF: 6878, 6881, 6882, 7869, 7870, 7871, 7872
- Grid Monitoring Service: 1976

  Note: Starting in SAS 9.4M6, Grid Management Services (GMS) is not updated to work with the latest release of LSF. Therefore, GMS is in Limited Support. See SAS Technical Support Services and Policies for more information about Limited Support.

- Platform Process Manager: 1966

If you need to change any port numbers, modify these files:

- LSF ports: LSF_ENVDIR/conf/lsf.conf and EGO_CONFDIR/ego.conf
- Grid Monitoring Service port: gms/conf/ga.conf

  Note: Starting in SAS 9.4M6, Grid Management Services (GMS) is not updated to work with the latest release of LSF. Therefore, GMS is in Limited Support. See SAS Technical Support Services and Policies for more information about Limited Support.

- Platform Process Manager port: pm/conf/js.conf

If you change the Grid Monitoring Service port, you must also change the metadata for the Grid Monitoring Server. If you change the Platform process Manager port, you must also change the metadata for the Job Scheduler Server.

Ports might be used by other programs. To check for ports that are in use, stop the LSF daemons and issue the command **`netstat -an`** |*search-tool port*, where *search-tool* is grep (UNIX) or findstr (Windows). Check the output of the command for the LSF ports. If a port is in use, reassign the port or stop the program that is using the port.

SAS assigns random ports for connections, but you can restrict the range of ports SAS uses by using the `-tcpportfirst` *`first-port`* and `-tcpportlast` *`last-port`* options. You can specify these options in the SAS configuration file or on the SAS command line. For remote sessions, you must specify these options either in the grid command script (sasgrid.cmd on Windows or sasgrid on UNIX) or in the **Command** field in the logical grid server definition in metadata. For example, adding the following parameters to the SAS command line in the grid script restricts the ports that the remote session uses to between 5000 and 5005:

```
-tcpportfirst 5000 -tcpportlast 5005
```

Verifying the Platform Suite for SAS Environment

Verifying That LSF Is Running

After the installation and configuration process is complete, verify that all of the LSF daemons are running on each machine.

For Windows machines, log on to each machine on the grid and check the Services dialog box to verify that these services are running:

- Platform LIM
- Platform RES
- Platform SBD

For UNIX machines, log on to each machine on the grid and execute the **`ps`** command to check for processes that are running in a subdirectory of the $LSF_install_dir. An example command follows:

```
ps -ef|grep LSF_install_dir
```

The daemons create log files that can help you debug problems. The log files are located in the machine's *LSF_install_dir*\logs directory (Windows) or the shared LSF_TOP/log directory (UNIX). If the daemon does not have access to the share on UNIX, the log files are located in the /tmp directory.

If the command fails, check the following:

- Verify that the path to the LSF programs is in the PATH environment variable. For LSF 7, the path is *LSF_install_dir*/7.0/bin.

- On UNIX machines, you might have to source the **LSF_TOP/conf/profile.lsf** file to set up the LSF environment.
- A machine might not be able to access the configuration files. Verify that the machine has access to the shared directory that contains the binary and configuration files, defined by the LSF_ENVDIR environment variable. If the file server that is sharing the drive starts after the grid machine that is trying to access the shared drive, the daemons on the machine might not start. Add the LSF_GETCONF_TIMES environment variable to the system environment and set the variable value to the number of times that you want the daemon to try accessing the share in each five-second interval before the daemon quits. For example, setting the variable to a value of 600 results in the node trying for 50 minutes ((600*5 seconds)/60 seconds per minute) before quitting.
- The license file might be invalid or missing. If LSF cannot find a license file, some daemons might not start or work correctly. Make sure that the license file exists, is properly referenced by the LSF_LICENSE_FILE parameter in the LSF_ENVDIR/conf/lsf.conf file, and is accessible by the daemons.
- All daemons might not be running. Restart the daemons on every machine on the grid using the **lsfrestart** command. If this command does not work, run the /etc/init.d/lsf restart command (UNIX) or use the Services Administration tool (Windows). Open Services Administration, stop the SBD, RES, and LIM services (in that order). Next, start the LIM, RES, and SBD services (in that order).
- A grid machine might not be able to connect to the SAS grid control machine. The grid control machine is the first machine listed in the lsf.cluster.*cluster_name* file. Make sure that the daemons are running on the master host, and verify that the machines can communicate with each other.

Verifying LSF Setup

You must verify that all grid machine names are specified correctly in the LSF_ENVDIR/conf/lsf.cluster.*cluster_name* file and the resource is specified in the lsf.shared file. Follow these steps to make sure the configuration is correct:

1. Log on as an LSF administrator on one of the machines on the grid, preferably the grid control server machine. The LSP Administrator ID is listed in the lsf.cluster.*cluster_name* file under the line
   `Administrators=username1username2 ... usernameN`.
2. Run the command **lsadmin ckconfig -v** to check the LSF configuration files for errors.
3. Run the command **badmin ckconfig -v** to check the batch configuration files for errors.
4. Run the command **lshosts** to list all the hosts in LSF, and to verify that all the hosts are listed with the proper resources.
5. Run the command **bhosts** to list all the hosts in LSF's batch system. Verify that all hosts are listed. Make sure that the Status for all hosts is set to **ok** and that the correct number of jobs slots have been defined for each host in the MAX column (the maximum number of jobs the host can process at the same time).

6 If you find any problems, correct the LSF configuration file and issue the commands **`lsadmin reconfig`** and **`badmin reconfig`** so that the daemons use the updated configuration files.

7 If you added or removed hosts from the grid, restart the master batch daemon by issuing the command **`badmin mbdrestart`**. To restart everything, issue the **`lsfrestart`** command.

Verifying LSF Job Execution

Some problems occur only when you run jobs on the grid. To minimize and isolate these problems, you can run debug jobs on specific machines on the grid.

To submit the debug job, run the command `bsub -I -m host_name set` from the grid client machine to each grid node. This command displays the environment for a job running on the remote machine and enables you to verify that a job runs on the machine.

If this job fails, run the `bhist -l job_id'` command, where *job_id* is the ID of the test job. The output of the command includes the user name of the person submitting the job, the submitted command, and all the problems LSF encountered when executing the job. Some messages in the bhist output for common problems include the following:

Failed to log on user with password
: specifies that the password in the Windows **`passwd.lsfuser`** file is invalid. Update the password using the **`lspasswd`** command.

Unable to determine user account for execution
: specifies that the user does not have an account on the destination machine. This condition can occur between a Windows grid client to a UNIX grid node because a domain has been prefixed to the Windows user name. Correct this problem by making sure that the user has an account on the UNIX machines. Also, add the line **`LSF_USER_DOMAIN=`** to the Windows lsf.conf file to strip the domain from the user name.

Verifying the SAS Environment

Verifying SAS Grid Metadata

SAS needs to retrieve metadata about the grid from a SAS Metadata Server in order to operate properly. Start SAS Management Console and use the Server Manager plug-in to verify the following:

Logical grid server
: Under the SAS Application Server context (for example, SASApp), verify that a logical grid server has been defined.

Open the Properties window for the logical grid server. Verify that the properties contain the correct path to the script file or the correct command that is executed on the grid node. Verify that the path exists on every node on the grid and that the command is valid on every node on the grid.

Grid monitoring server
: Verify that a grid monitoring server has been defined.

  Open the connection properties for the server. Verify that the properties contain the name or address of the machine that is running the Grid Monitoring Server daemon (typically the SAS grid control machine). Verify that the port specified in the properties is the same as that specified in the Grid Monitoring Service configuration file (the default value is 1976).

Verifying Grid Monitoring

The Grid Manager plug-in for SAS Management Console displays information about the grid's jobs, hosts, and queues. After you define the Grid Monitoring Server and the Grid Management Service is running on the control server, grid information is displayed in the Grid Manager plug-in in SAS Management Console. Common error messages encountered in the Grid Manager plug-in include the following:

Connection timed out or Connection refused
: The Grid Management Service is not running. Start the Grid Management Service on the grid control machine.

Your user ID or password is invalid. Please try again or contact your systems administrator
: Either the user provided invalid credentials for the machine running the Grid Monitoring Service, or the user's credentials that are stored in the metadata do not include a password for the login that is associated with the authorization domain used by the Grid Monitoring Server connection. For example, "Grid 1 Monitoring Server" is defined in the metadata to use the "DefaultAuth" authorization domain. A login has been defined for "User1" in the User Manager for the "DefaultAuth" domain, but only the user ID has been specified and the password is blank.

  There are three ways to correct the problem. First, provide complete credentials for the authorization domain for the user. Second, you can remove the login for the authorization domain. The third option is to use a different authorization domain for the Grid Monitoring Server connection. If you provide the correct credentials, the user is not prompted for a user ID and password. If you remove the login for that authorization domain or change the Grid Monitoring Server connection to use a different authorization domain without adding credentials for the user for that domain, the user is prompted for their user ID and password to connect to the machine where the Grid Monitoring Server is running.

Verifying SAS Job Execution

SAS provides a grid test program on the SAS support website that tests connectivity to all nodes in a SAS Grid Manager for Platform grid. Run the program from a grid

client. You can download the program from SAS Grid Computing: Code to test initial installation. After you download the program, follow these steps:

1. Copy and paste the grid test program into a SAS Foundation Display Manager session.
2. If the application server associated with your logical grid server in your metadata is not named “SASMain”, change all occurrences of “SASMain” in the test program to the name of the application server that is associated with your logical grid server. For example, some SAS installations have named the application server “SASApp”, so all occurrences of SASMain should be replaced with “SASApp”.
3. Submit the code.

The program attempts to start one remote SAS session for every job slot available on the grid. The program might start more than one job on multi-processor machines, because LSF assigns one job slot for each core by default.

Here are some problems that you might encounter when running the grid test program:

`Grid Manager not licensed` message
: Make sure that your SID contains a license for SAS Grid Manager.

`Grid Manager cannot be loaded` message
: Make sure that Platform Suite for SAS has been installed and that the LSF and PATH environment variables are defined properly.

`Invalid resource requested` message
: The application server name or workload value has not been defined in the lsf.shared file. Also, make sure that you associate the value with the hosts that you want to run SAS programs in the lsf.cluster.*cluster_name* file.

The number of grid nodes is 0.
: Possible reasons for this error include the following:

- The application server name was not defined as a resource name in the lsf.shared file.
- The application server name was not associated with any grid nodes in the lsf.cluster.*cluster_name* file.
- The grid client where the job was submitted cannot communicate with the entire grid.

The number of grid nodes is not the same as the number of grid node machines.
: As shipped, the number of grid nodes equals the number of job slots on the grid. By default, the number of job slots is equal to the number of cores, but the number of job slots for a grid node can be changed.

  Another explanation is that the application server name has not been associated with all the grid nodes in the lsf.cluster.*cluster_name* file.

Jobs fail to start.
: Possible reasons for this problem include the following:

- The grid command defined in the logical grid server metadata is either not valid on grid nodes or does not bring up SAS on the grid node when the command is run. To verify the command, log on to a grid node and run the command defined in the logical grid server definition. The command should attempt to start a SAS session on the grid node. However, the SAS session might not run successfully because grid parameters have not been included.

Platform Suite for SAS provides a return code of 127 if the command to be executed is not found and a return code of 128 return code if the command is found, but there is a problem executing the command.

- Incorrect version of SAS installed on grid nodes. SAS 9.1.3 Service Pack 3 is the minimum supported version. A return code of 231 might be associated with this problem.
- Unable to communicate between the grid client and grid nodes. Verify that the network is set up properly, using the information in "Verifying the Network Setup " on page 168.

Jobs run on machines that are supposed to be only grid clients.

By default, all machines that are listed in the lsf.cluster.*cluster_name* file are part of the grid and can process jobs. If you want a machine to be able to submit jobs to the grid (a grid client) but not be a machine that can process the job (a grid node), set its maximum job slots to 0 or use the Grid Manager plug-in to close the host.

Debugging Grid Jobs

When a grid-enabled SIGNON is used or a job is submitted by SASGSUB, the sasgrid script is used to start a SAS session on the grid node. To aid in debugging, the output from the script (but not the SAS log and SAS output) is written to a unique log file. The file is located in the `GridServer/Logs` directory of the machine where the job runs. If the job is successful, the log file is deleted when the job finishes running. If the job is unsuccessful, the log file is saved.

The log file is named using the format SASGrid.*userID*.*hostname*.*process_ID*.*YYYMMDDhhmmss*.log

The log file is not created if the job is run on a grid-enabled IOM server (workspace server, stored process server, or pooled workspace server), because these servers do not use the sasgrid script. These servers are started by the object spawner, so you must configure logging of the object spawner to debug grid-enabled IOM servers. For example, you can set the logger App.tk.tkegrid to TRACE and then review the object spawner's log.

Turning On Debugging in the SAS Environment Manager Agent Plug-In Loggers

To turn on debugging for the SAS Environment Manager agent plug-in loggers, add the following lines to the file `levelroot/Web/SASEnvironmentManager/agent-5.8.0-EE/conf/agent.properties`:

```
log4j.logger.com.sas.grid.hyperic=DEBUG
log4j.logger.com.sas.svcs.security.authentication.client=DEBUG
```

Note: As of release 9.4M8, Log4J2.default release 2.17.1 is used.

Fixing Issues with the SAS Grid Manager Module for SAS Environment Manager

Here are some possible solutions if the SAS Grid Manager module fails to invoke in SAS Environment Manager:

You receive a “Service is temporarily not available” error

Verify that Platform Web Services is running. Restart the service if needed.

You receive a “Connecting to the server failed” error

Verify that SAS Environment Manager is running.

You receive an “Internal server error” message

Verify that Platform LSF is running.

You receive the message "You do not have permission to perform this action" when attempting to use the SAS Grid Manager module on Windows

This message indicates that Platform Web Services is not running under the LSF Administrator account. To change the account that the service is running under, follow these steps:

1. In the Windows Services window, click the entry for the service **SASServer14_1** and select **Stop** from the context menu.
2. In the dialog box for the SASServer14_1 service, click the **Log On** tab.
3. Select the **This account** radio button and specify the user ID and password of the LSF Administrator user. Click **OK** to close the dialog box.
4. Click the entry for the service and select **Start** from the context menu.

If you would rather issue commands, open a Windows Command window and issue these commands:

- net stop "SAS [*SASConfig-Lev*] SASServer14_1 - WebAppServer"
- sc config "SAS [*SASConfig-Lev*] SASServer14_1 - WebAppServer" obj= *domain* *userid* password= *password*

  Note: You must include a space between the equal sign and the value for the obj= and password= parameters.

- net start "SAS [*SASConfig-Lev*] SASServer14_1 - WebAppServer"

Substitute your directory names for *SASConfig-Lev*.

You must update the entry for this service anytime the password for the LSF Administrator account changes.

You receive permission errors or LSF library initialization errors

Verify that Platform Web Services is running under the LSF Administrator account. These errors occur if Platform Web Services runs under a local user account. To change the account that the service is running under, follow these steps:

1. In the Windows Services window, click on the entry for the service **SASServer14_1** and select **Stop** from the context menu.
2. In the dialog box for the SASServer14_1 service, click the **Log On** tab.
3. Select the **This account** radio button and specify the user ID and password of the LSF Administrator user. Click **OK** to close the dialog box.
4. Click the entry for the service and select **Start** from the context menu.

Examining the debug log might help diagnose problems with the Grid Manager plug-in. To start logging, complete the following steps:

1. Depending on your release, one of the following files is available. Open it.
   - server-log4j.xml located in the directory `$SASEV/conf`
   - SASGridManager-log4j.xml located in the directory `$SASConfig/LevN/Web/Common/LogConfig`
2. Change the root-level value to `DEBUG`.

Depending on the release, the debugging information is written to the file server.log in one of the following directories:

- `$SASEV/logs`
- `$SASConfig/LevN/Web/Logs/SASServer14_1`

Note: As of release 9.4M8, Log4J2.default release 2.17.1 is used.

PART 4

Grid Computing for SAS Using SAS Grid Manager for Hadoop

11

Using SAS Grid Manager for Hadoop

Overview

SAS Grid Manager for Hadoop provides workload management, accelerated processing, and scheduling of SAS analytics co-located on your Hadoop cluster. SAS Grid Manager for Hadoop leverages YARN to manage resources and distribute SAS analytics to a Hadoop cluster running multiple applications. Integration with Oozie provides the scheduling capability for SAS workflows. All of the existing SAS Grid syntax, submission modes, and integration with other SAS products and solutions is supported by SAS Grid Manager for Hadoop. Therefore, end users of SAS applications that are submitting work to the SAS grid will not notice any difference in their operations. They will continue to submit jobs as they normally would, but those jobs will be processed on the Hadoop cluster. Grid monitoring and management is provided through the Hadoop monitoring tools, rather than through SAS Environment Manager or SAS Management Console.

Processing in the cluster is controlled by the queue allocations defined in the YARN scheduler and through the SAS Grid Manager for Hadoop's SASGRID policy file. The queues determine the allocation of resources between different groups. The SASGRID policy file enables you to specify which hosts in the Hadoop cluster are available for grid jobs. You can also define the application types (such as normal or priority) that are processed by the grid, specify which hosts can be used for processing those applications, and specify the computing resources and limits (such as memory and maximum processing time) that apply to each type. You can then specify application type to associate with a submitted job in a number of ways, including using a grid option on a grid options set. When a SAS user submits a job, the correct grid options set, including the application type, is applied and the job is processed on the designated hosts in the Hadoop cluster.

Scheduling of SAS jobs on the Hadoop cluster is provided through Oozie. SAS uses Oozie to schedule DATA step batch servers to process SAS code in workflows. SAS users do not interface directly with Oozie. They deploy jobs for scheduling and schedule those jobs using an Oozie scheduling server (defined in SAS metadata), just as they would with any other type of scheduling. The Oozie scheduling server definition contains the information needed to send the scheduled jobs to Oozie on the Hadoop cluster.

High availability of the Resource Manager and the Name Node are supported by SAS Grid Manager for Hadoop. In addition, any failed SAS job is resubmitted.

Configuration

SAS Grid Manager for Hadoop supports these Hadoop distributions:

- Cloudera 5.2.x and later
- Hortonworks 2.1.x and later
- Hortonworks 2.2 (this version includes a REST API for job submission that results in better performance)
- MapR 4.0.x and later
- MapR 4.1.x and later (recommended, because this version adds support for crontab, which is required in order to schedule recurring flows)

The process of configuring SAS Grid Manager for Hadoop consists of these steps:

1. Pre-install steps. These are discussed in Chapter 2 of Configuring the Hadoop Cluster for Use by SAS® Grid Manager for Hadoop.
   a. Set up a shared file system.
   b. Set up users.
   c. Install Hadoop services.
   d. Enable Kerberos on the Hadoop cluster.
   e. Enable SSL.
   f. Update YARN parameters.
   g. Set up HDFS directories.

h Run a test mapreduce job to verify that all parameters and services have been configured correctly..

2 Run the SAS Deployment Wizard to install and configure a SAS Grid Manager for Hadoop control server.

3 Post-installation steps. These are discussed in Chapter 4 of Configuring the Hadoop Cluster for Use by SAS® Grid Manager for Hadoop.

a Create a grid policy file. See "Creating the SASGRID Policy File" on page 200.

b Update the Java policy files, if needed for Kerberos encryption.

c Verify the object spawner keytab. See "Using Kerberos Authentication with SAS Grid Manager for Hadoop" on page 200.

d Extract the SSL certificates.

e If you are using a MapR cluster, install the MapR client.

f Deploy SAS to the grid nodes (if you are not installing on a shared file system).

g Set up scheduling directories.

Note: In order to run Hadoop in-database jobs in a SAS grid environment, you must change the HADOOP_TOKEN_FILE_LOCATION environment variable. Open the file ***grid-application-server-context-path/*** **appservercontext_env_usermods.sh** and add the line `unset HADOOP_TOKEN_FILE_LOCATION`. Save and close the file.

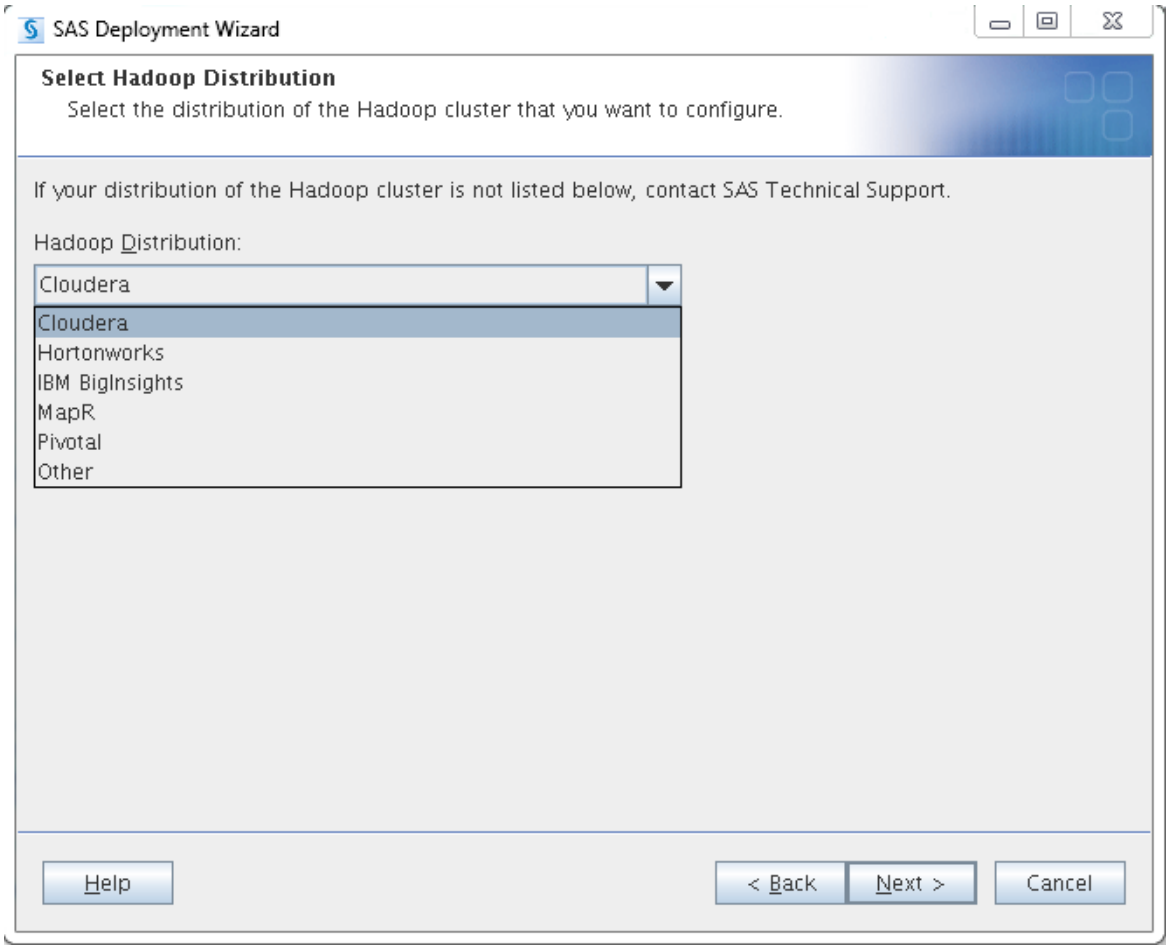
Using SAS Deployment Wizard to Deploy SAS Grid Manager for Hadoop

After you have completed the pre-installation process that is described in Configuring the Hadoop Cluster for Use by SAS® Grid Manager for Hadoop, use the SAS Deployment Wizard to deploy SAS Grid Manager for Hadoop. Follow these steps:

1 Log on as the SAS install user to the machine that contains the Resource Manager.

2 Run the command `kinit` to initialize the Kerberos credential cache.

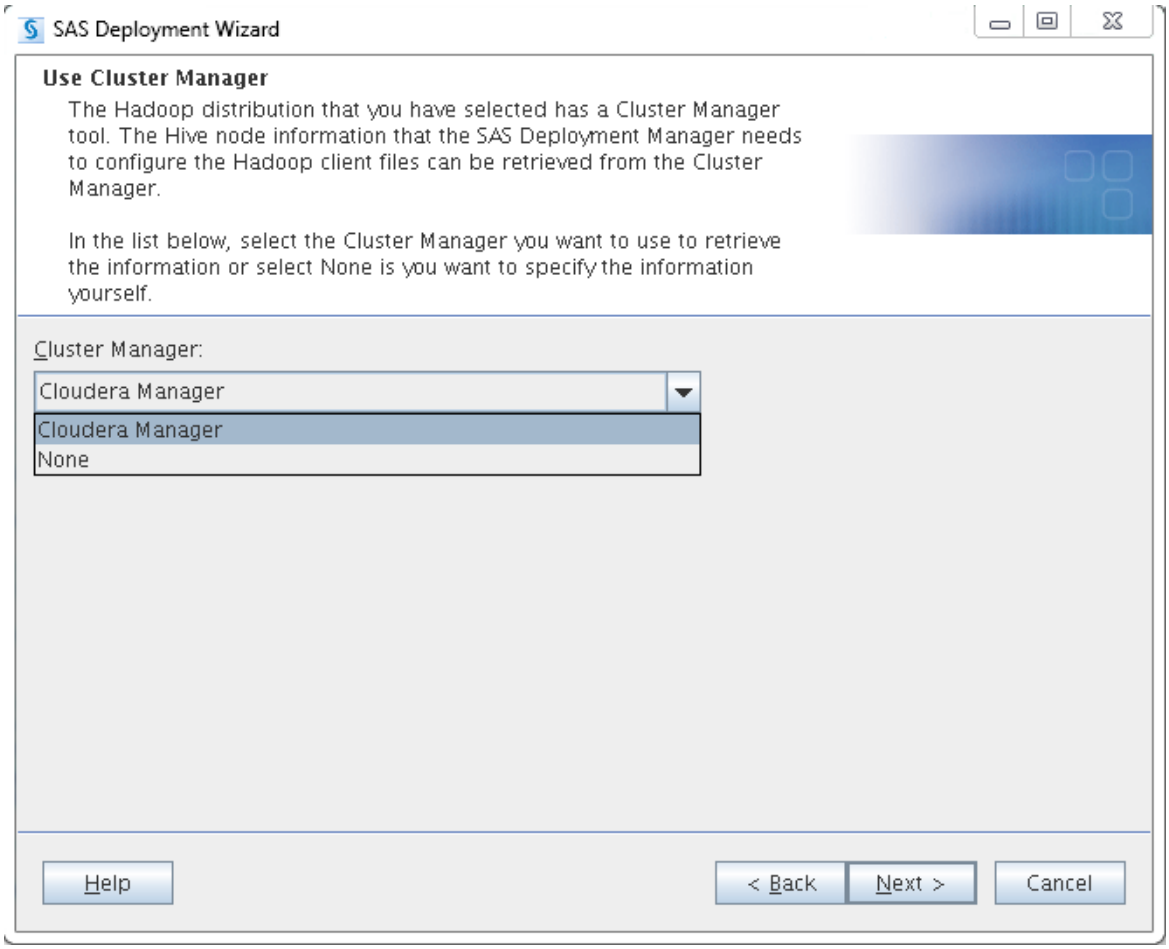
3 Start the SAS Deployment Wizard. The Select Hadoop Distribution page opens.



4 From the drop-down menu, select the Hadoop distribution that you want to configure.

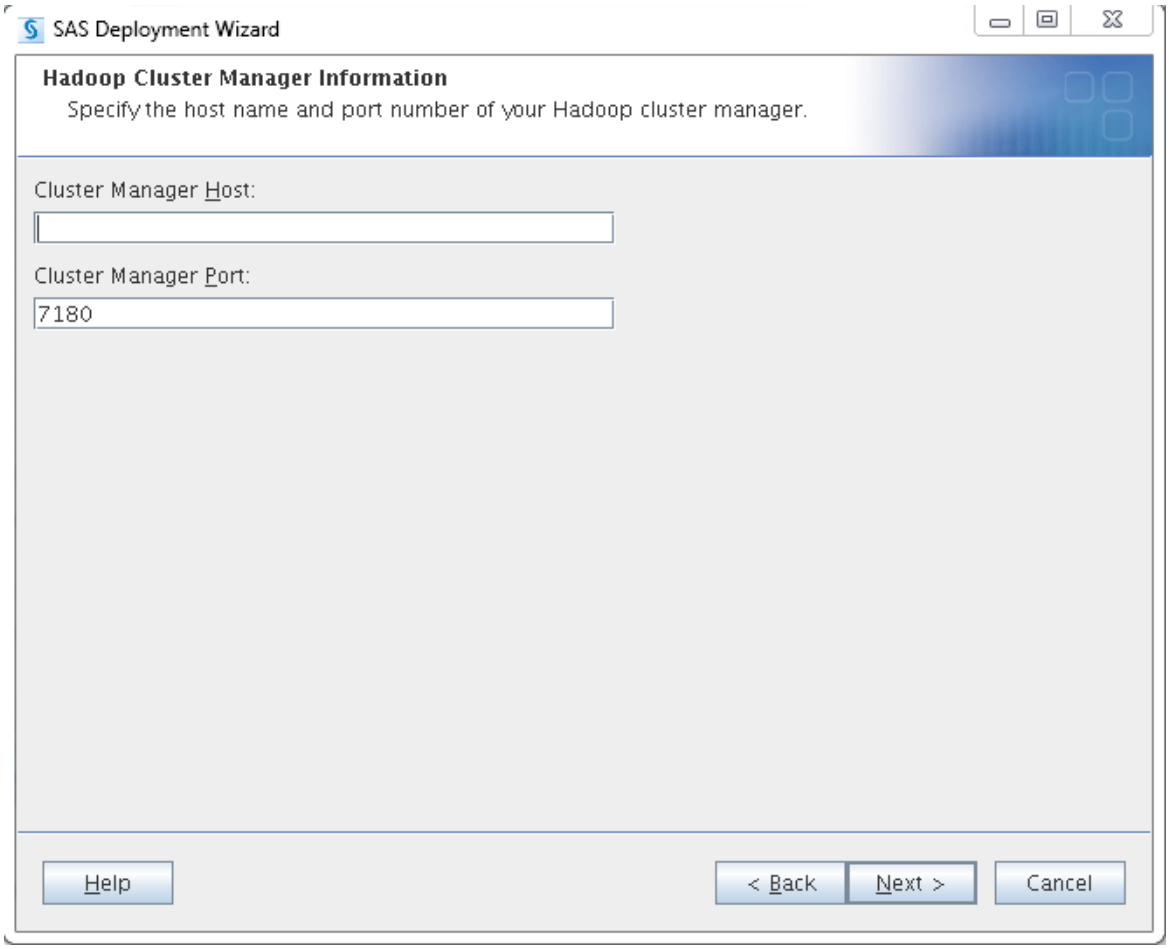
Note: If your distribution is not listed, exit the SAS Deployment Manager and contact SAS Technical Support.

Click **Next**. The Use Cluster Manager page opens.



5 Select the cluster manager from the drop-down list. If you specify the cluster manager, the SAS Deployment Wizard can retrieve information about the Hive node for use in configuring the Hadoop clients. If you want to specify this information manually, select **None** in this field.

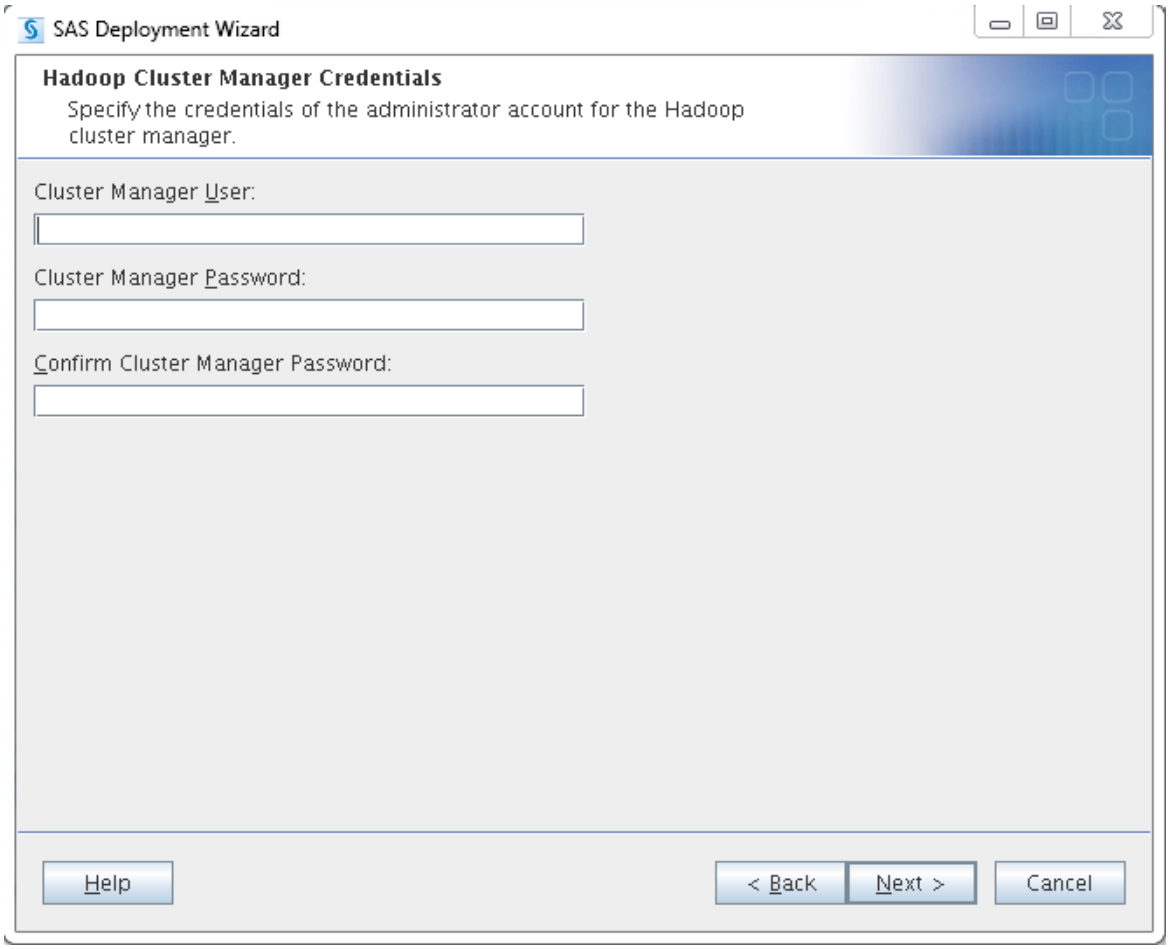
Click **Next**. The Hadoop Cluster Manager Information page opens.



6 Enter the host name and port number for your Hadoop cluster. For Cloudera, enter the location where Cloudera Manager is running. For Hortonworks, enter the location where the Ambari server is running. The port number is set to the appropriate default after Cloudera or Hortonworks is selected.

Note: The host name must be a fully qualified domain name. The port number must be valid, and the cluster manager must be listening.

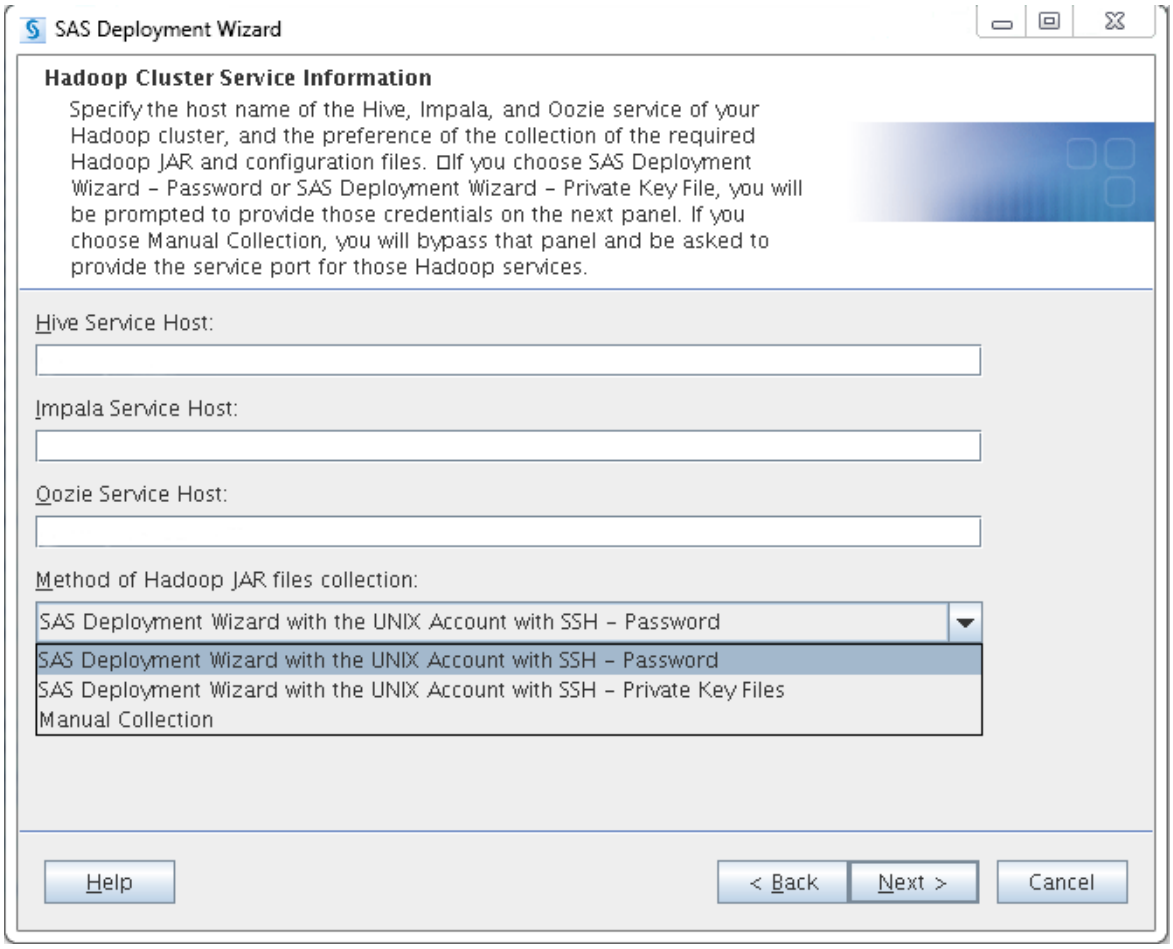
Click **Next**. The Hadoop Cluster Manager Credentials page opens.



7 Enter the Cloudera Manager or Ambari administrator account name and password.

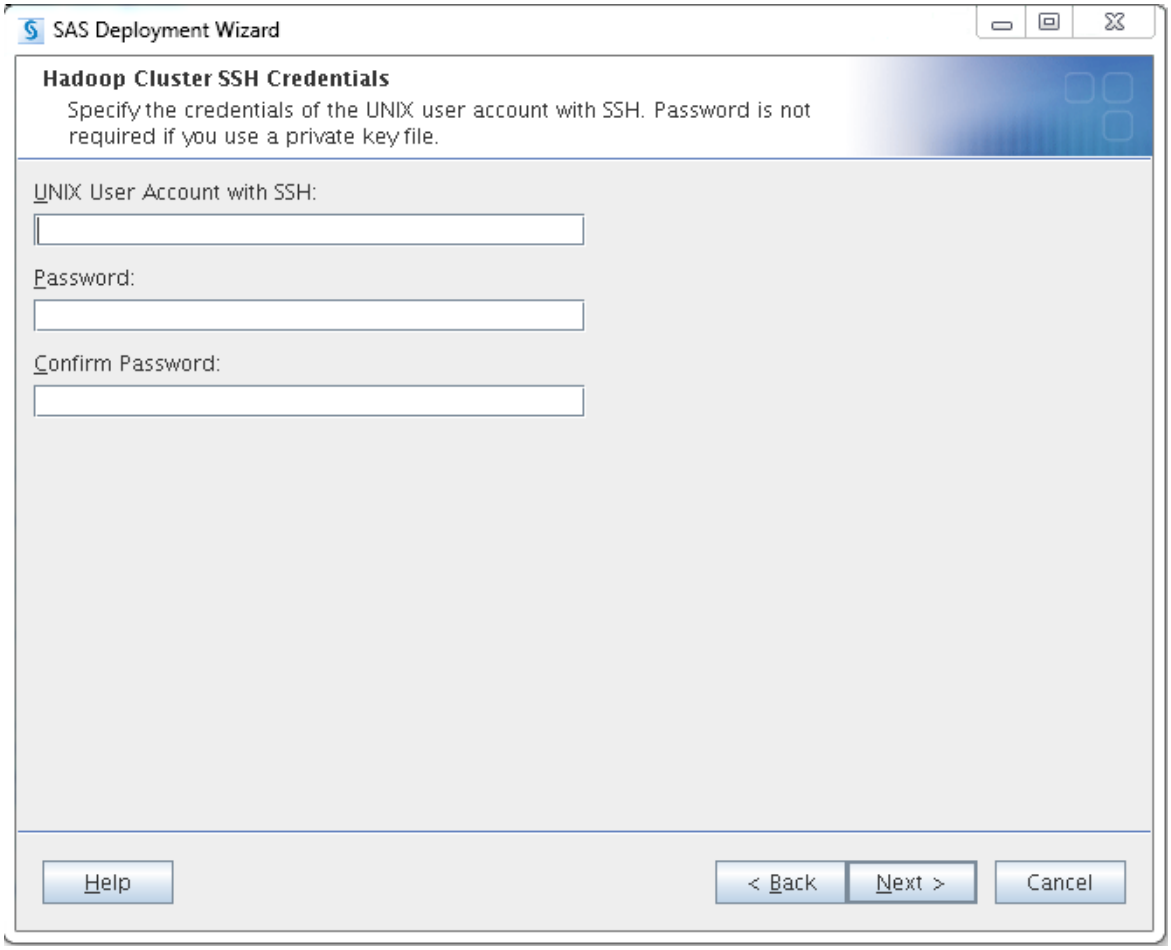
Note: Using the credentials of the administrator account to query the Hadoop cluster and to find the Hive node eliminates guesswork and removes the chance of a configuration error. However, the account name does not have to be that of an administrator; it can be a read-only user.

Click **Next**. The Hadoop Cluster Service Information page opens.



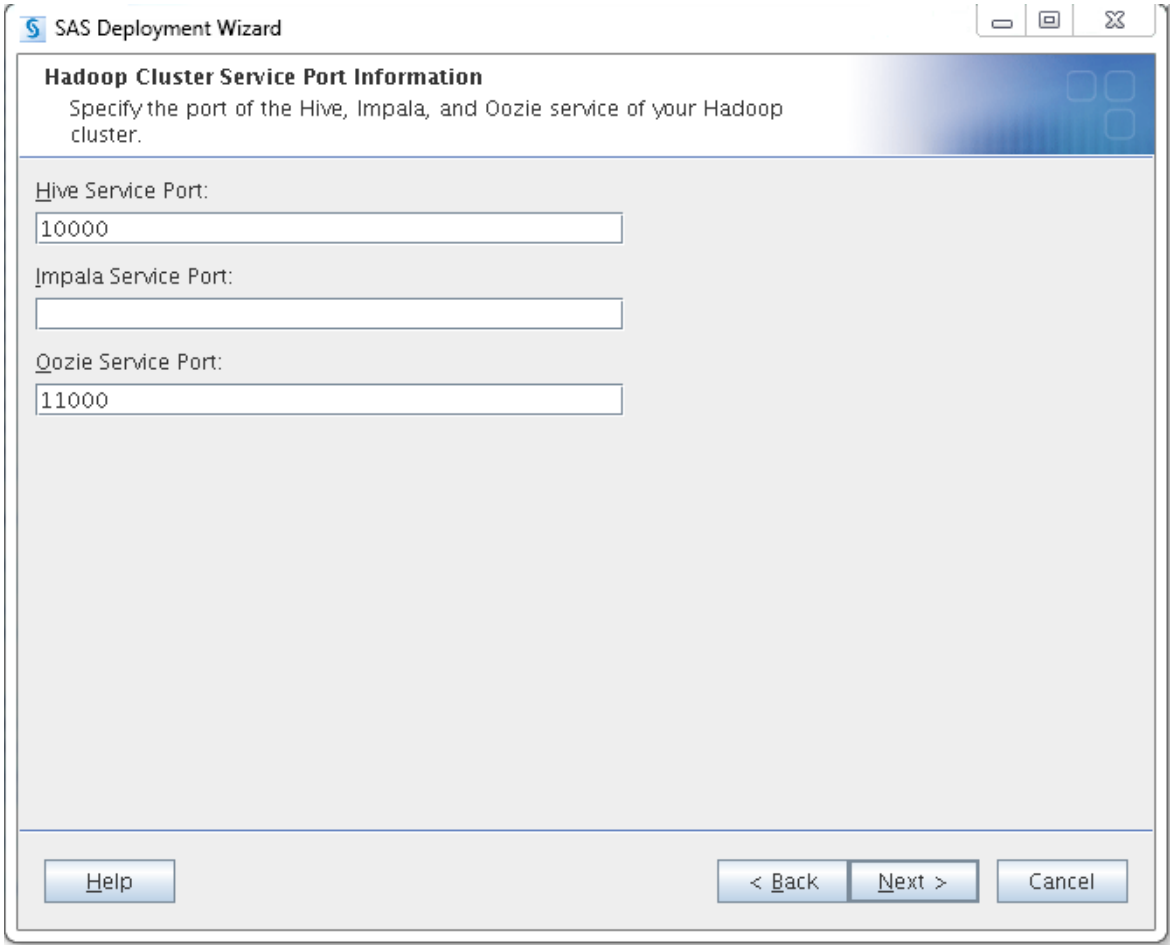
8 Specify the host names for the Hive, Impala, and Oozie services for your Hadoop cluster. Also specify the method that should be used to collect the Hadoop configuration and JAR files.

Click **Next**. The Hadoop Cluster SSH Credentials page opens.



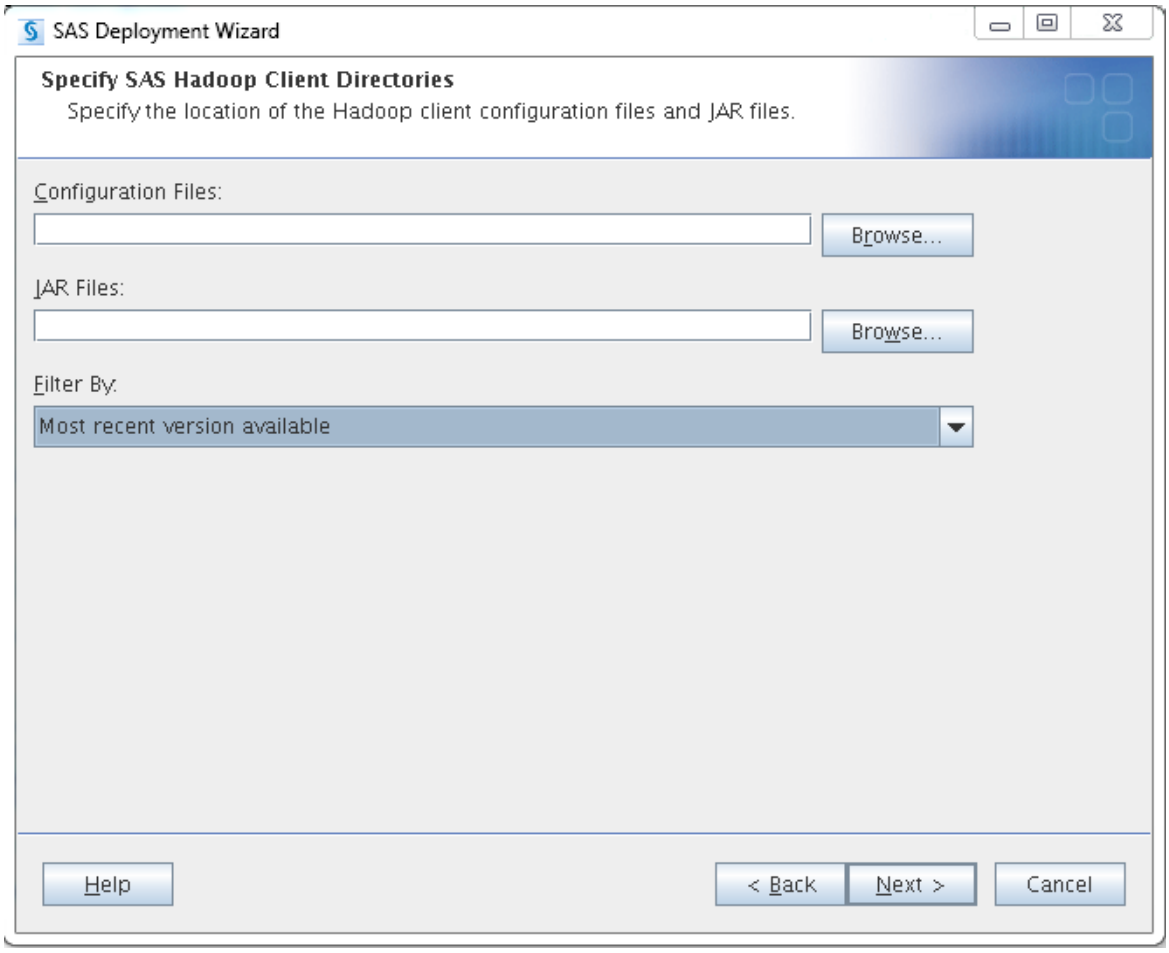
9 Specify the credentials for the account with SSH that is used to connect to the Hadoop cluster.

Click **Next**. The Hadoop Cluster Service Port Information page opens.



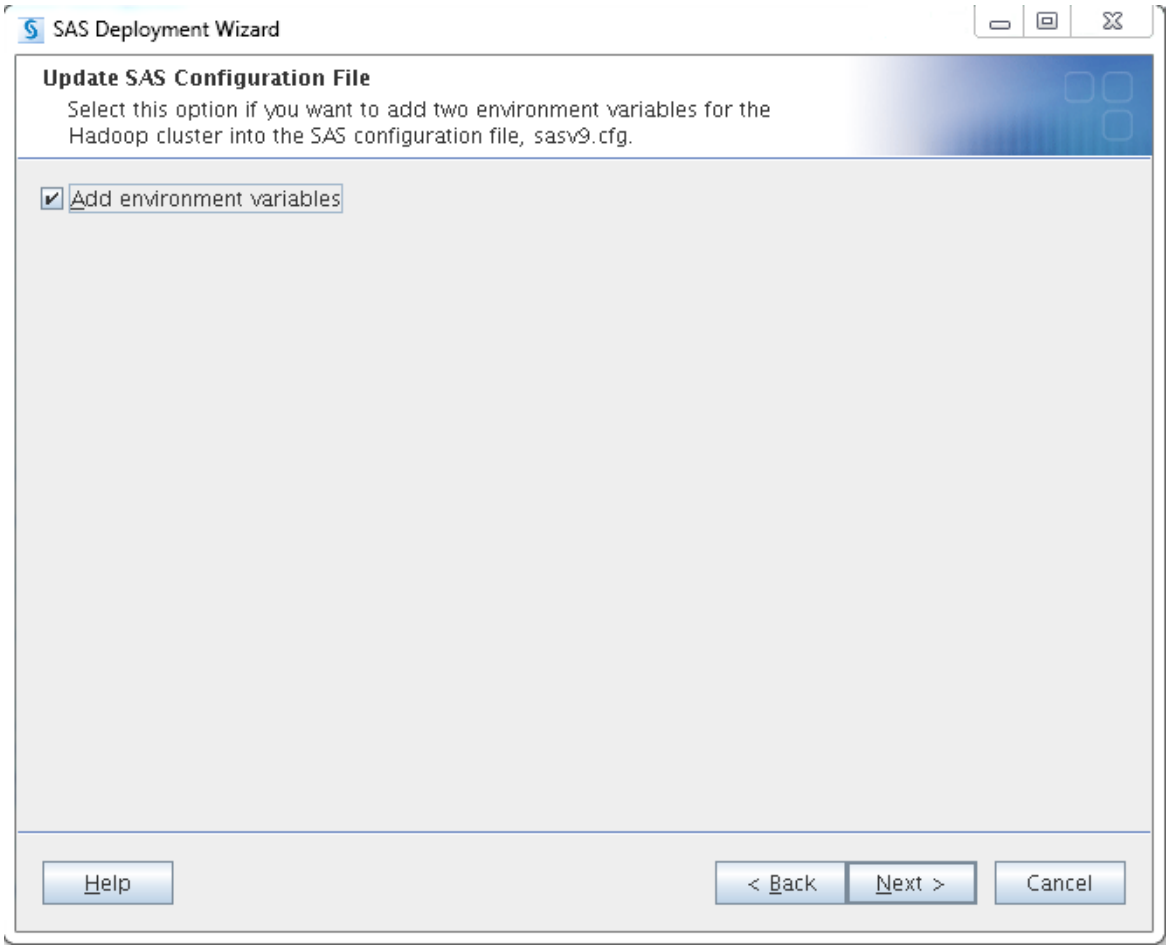
10 Specify the ports used by the Hive, Impala, and Oozie services of your Hadoop cluster.

Click **Next**. The Specify SAS Hadoop Client Directories page opens.



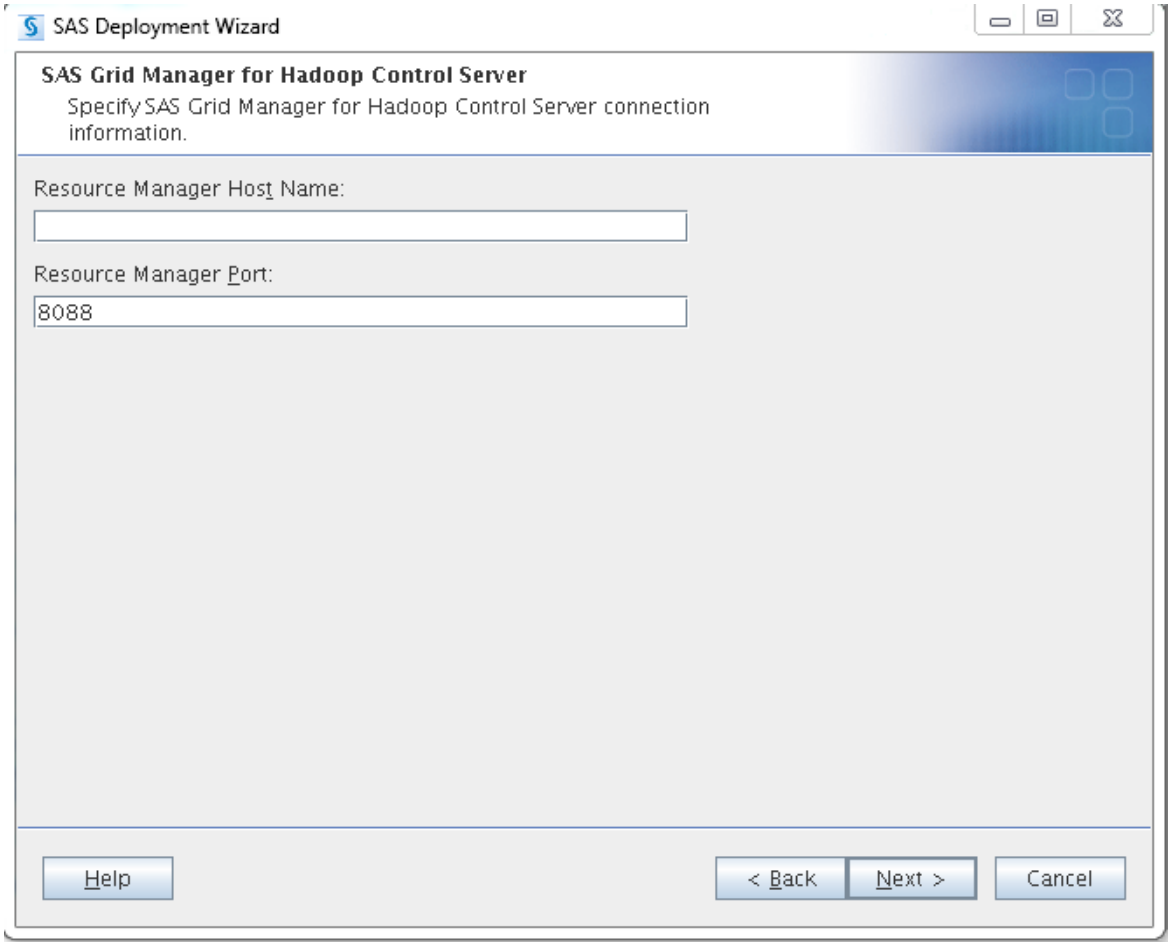
11 Specify the locations of the Hadoop configuration files and the Hadoop distribution JAR files.

Click **Next**. The Update SAS Configuration File page opens



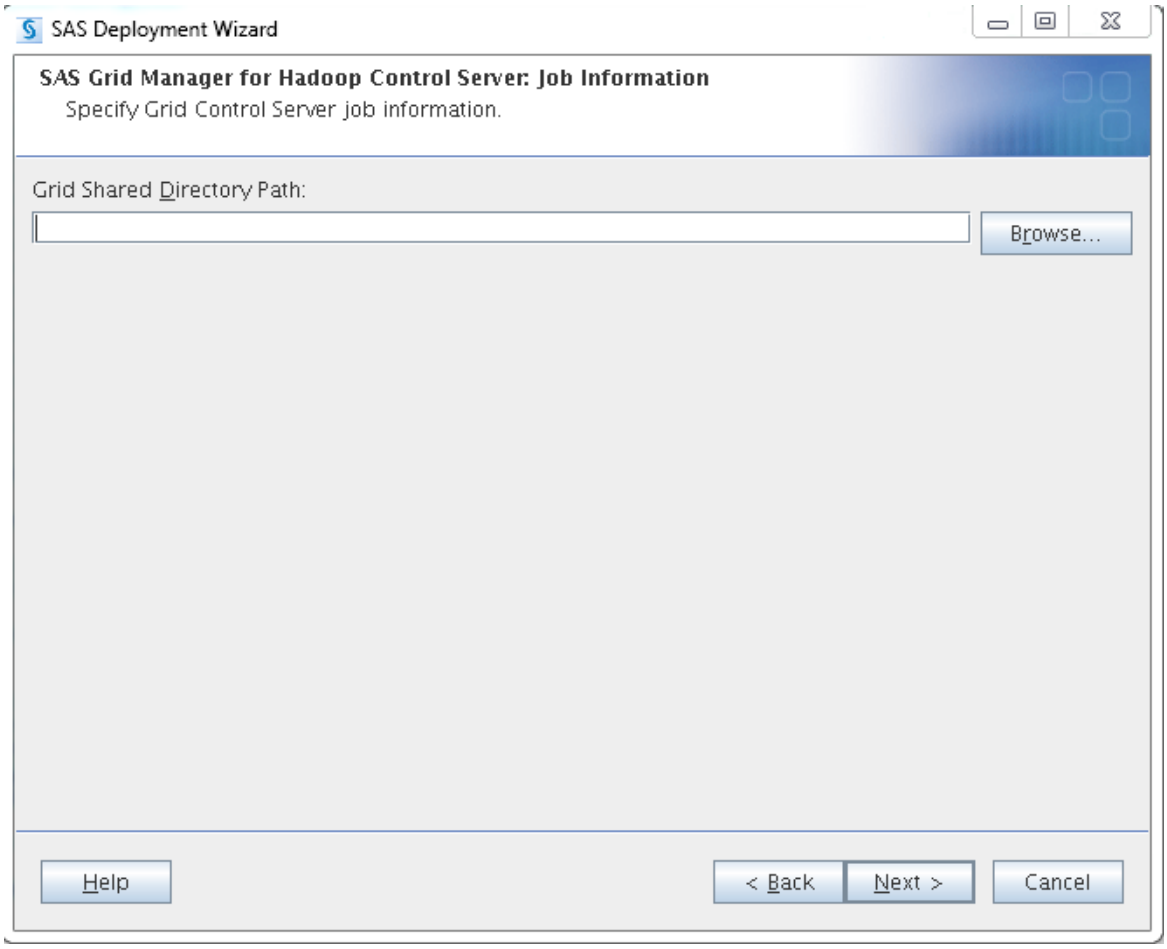
12 Select **Add environment variables** to add the locations of the Hadoop configuration file and distribution JAR files (specified in the previous step) to the SAS configuration file.

Click **Next**. The SAS Grid Manager for Hadoop Control Server page opens.



13 Specify the host name and port of the resource manager.

Click **Next**. The SAS Grid Manager for Hadoop Control Server: Job Information page opens



14 Specify the path of the directory shared by all hosts on the grid.

Grid Command
the script, application, or service that SAS uses to start server sessions on the grid nodes. Any SAS options that are included in this command are passed to the grid jobs. You can specify any additional SAS options in the **Grid Options** field.

This value is the path to the sasgrid script file (UNIX). Because this same command is used to start the servers on all grid nodes, the path to the directory on each grid node must be the same. For example: `/SAS/Grid/Lev1/SASApp/GridServer/sasgrid`.

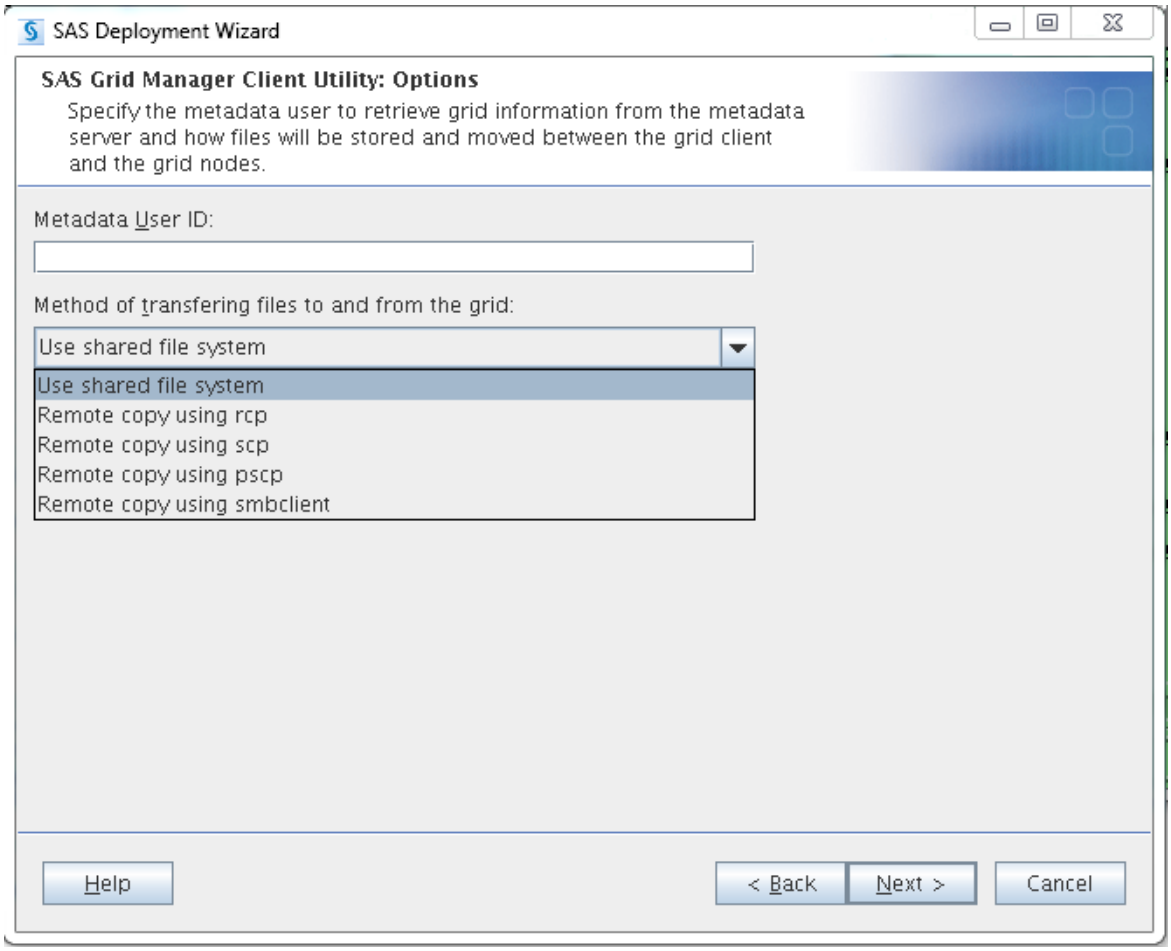
Grid Options
specifies the grid command options that are applied by default to jobs that are sent to the grid. Examples include the job priority, the job queue, or the user group that is associated with the job. Any grid options that are specified by a grid options set and mapped to a specific application override this value. Job options are specified as name/value pairs in this format. Multiple pairs are separated by semicolons

Workload(s) defined in grid
specifies the optional resources that can be processed on the grid.

Grid Provider Module
specifies the module other than the default used to connect to the grid. The module name is normally the default module used by the grid provider, so in most cases this value is eft blank.

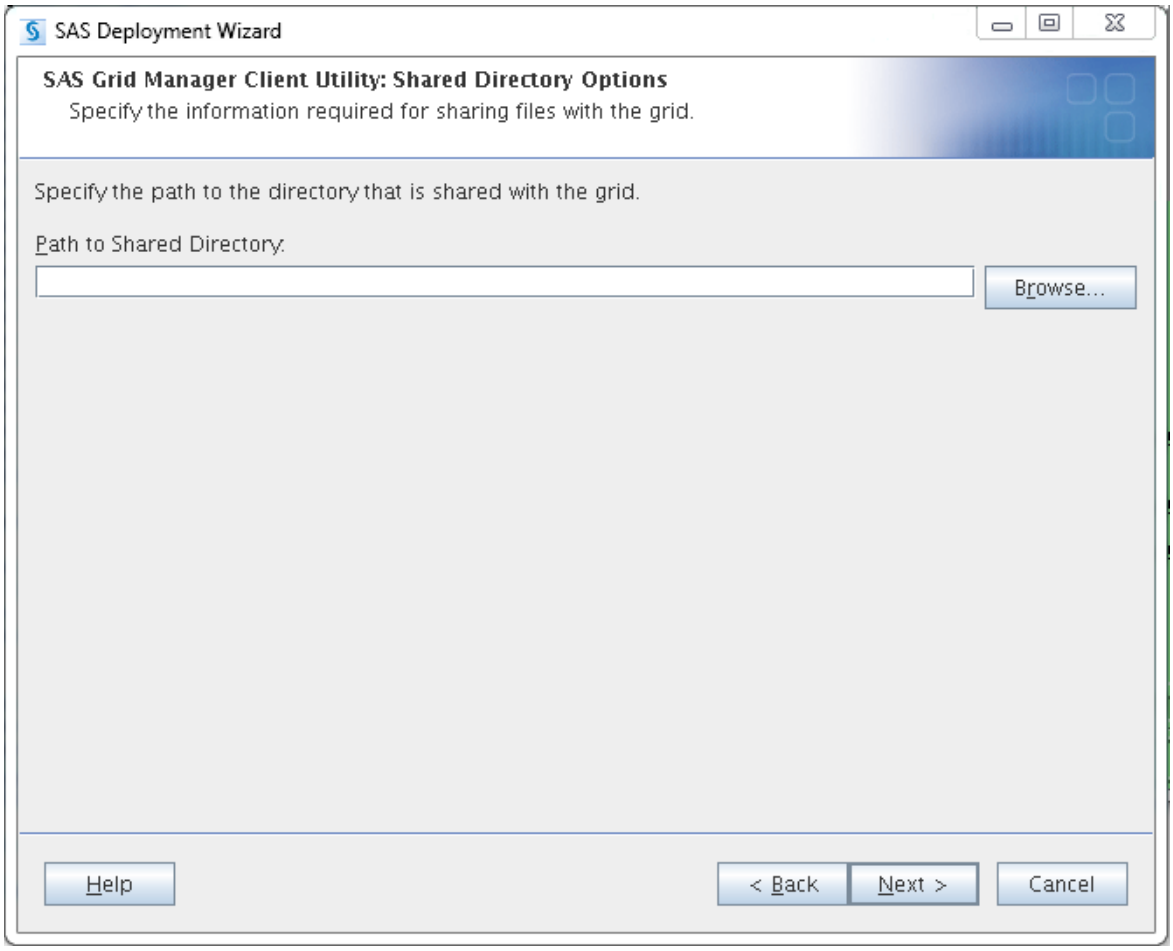
Grid Shared Directory Path
specifies the path of the directory shared by all hosts on the grid. This directory is used to store job information for the SAS Grid Manager Client Utility.

Click **Next**. The SAS Grid Manager Client Utility: Options page opens



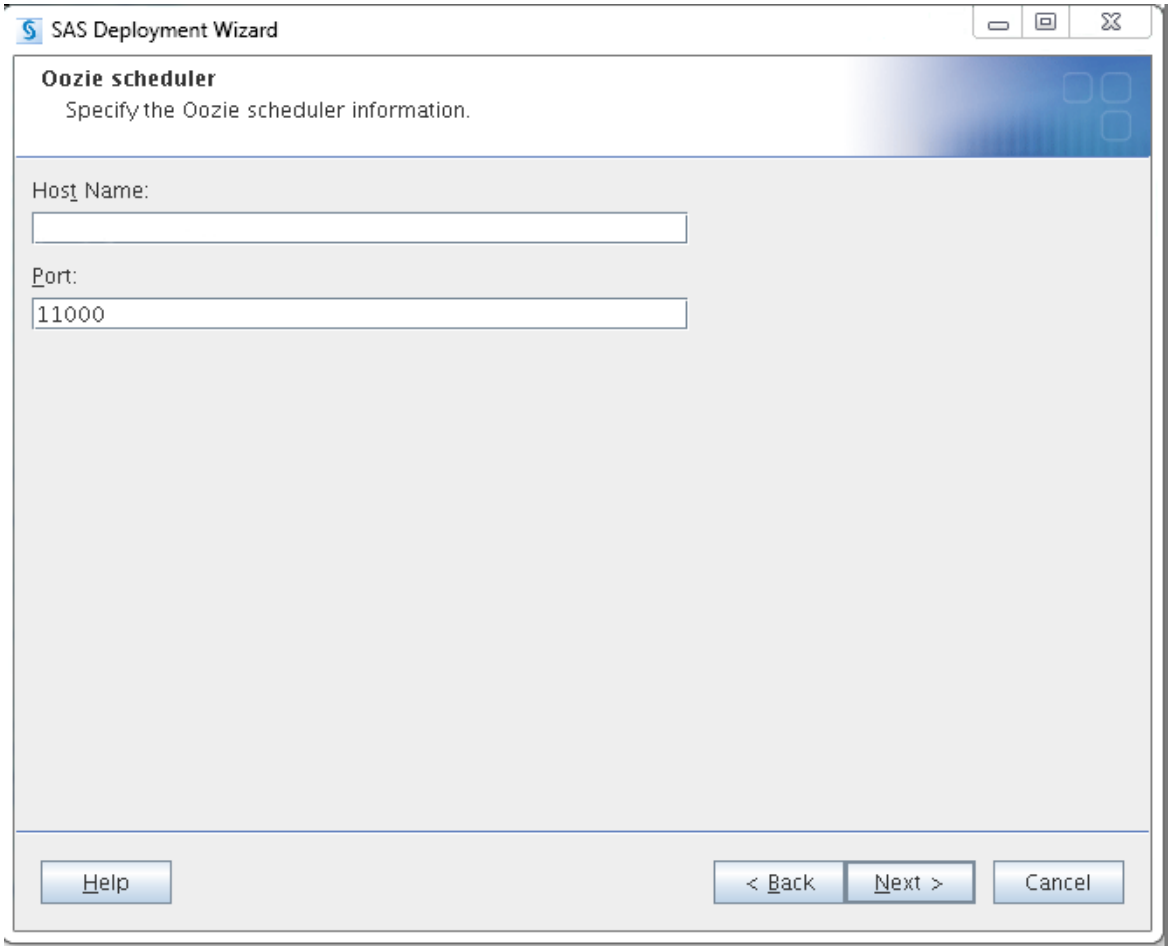
Specify the user ID used to connect to the SAS Metadata Server and the method used to transfer files between the grid client and the grid nodes.

15 Click **Next**. The SAS Grid Manager Client Utility: Shared Directory Options page opens



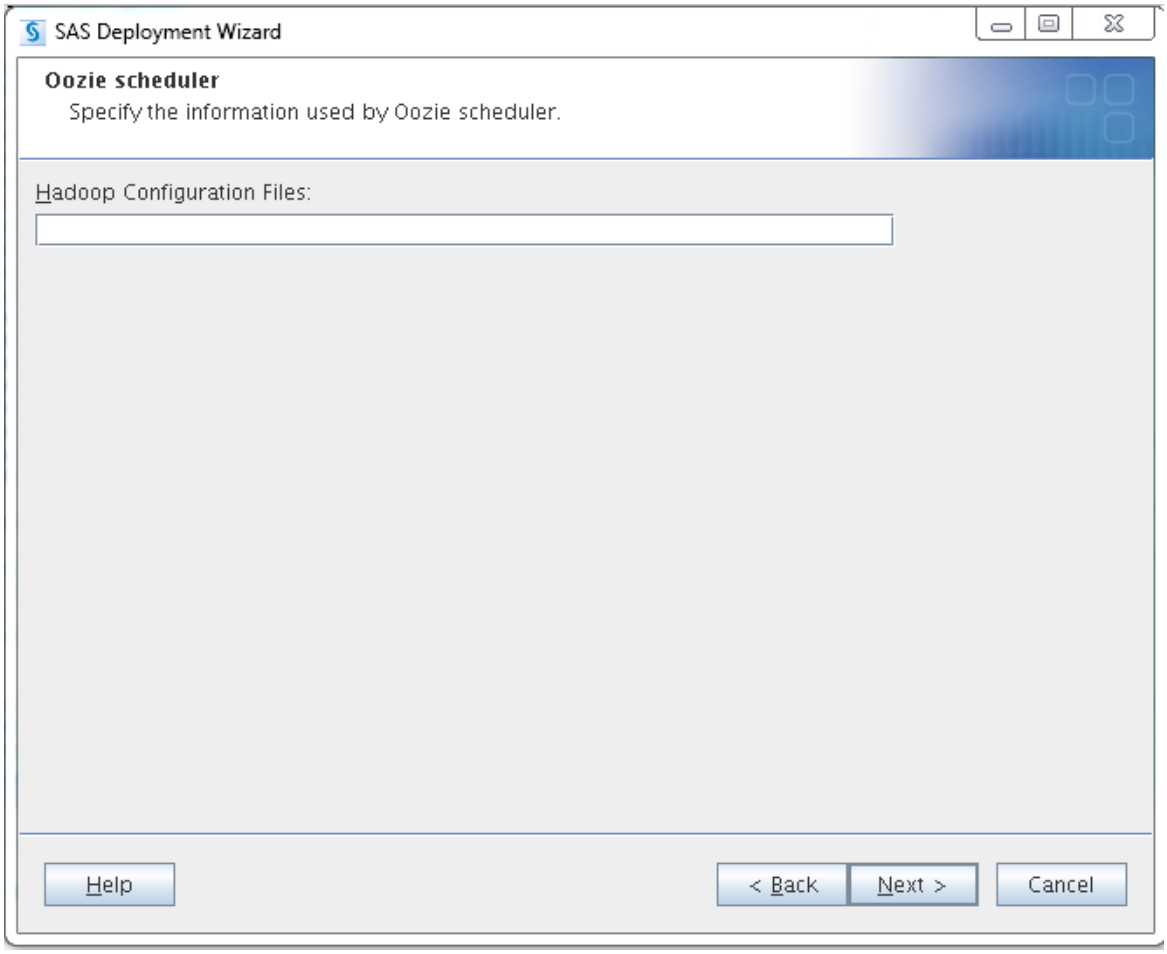
Specify the path of the directory shared by all hosts on the grid. This directory is used to store job information for the SAS Grid Manager Client Utility.

16 Click **Next**. The Oozie scheduler page opens



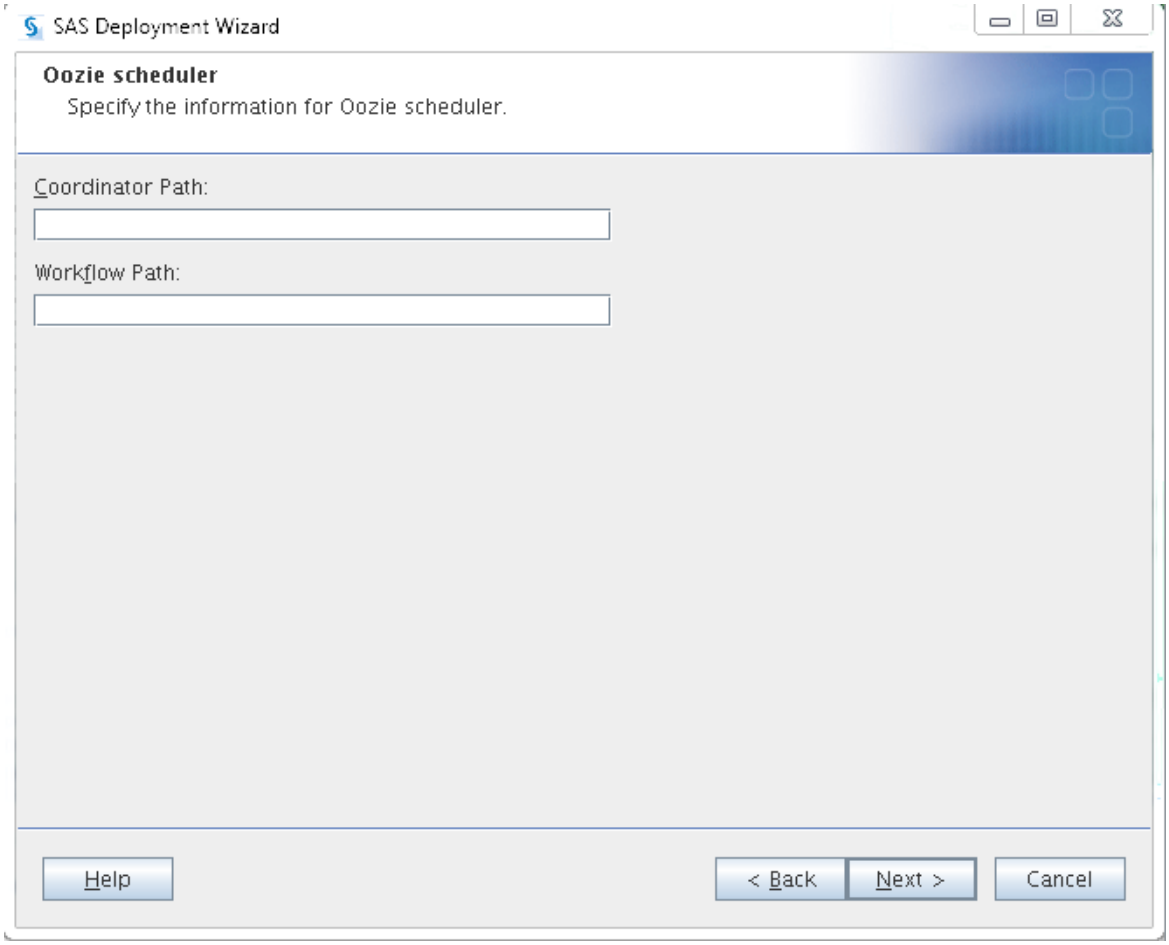
Specify the host name and port of the Oozie scheduling server.

17 Click **Next**. The Oozie scheduler page opens



Specify the location of the Hadoop configuration files.

18 Click **Next**. The Oozie scheduler page opens



Specify the location for the Oozie coordinators.

19 Click **Next**. The Deployment Summary page appears, which lists all of the software that will be installed and configured. Click **Start** to begin installing the SAS Grid Manager for Hadoop components.

Note: It takes several minutes to complete the configuration.

If the configuration is successful, the page title changes to Deployment Complete.

If warnings or errors occur, fix the issues and restart the configuration.

20 Click **Next** to close the SAS Deployment Wizard.

Using Kerberos Authentication with SAS Grid Manager for Hadoop

If your deployment is using Kerberos authentication, you must change a configuration parameter and the objspawn.keytab file. These changes enable ticket-granting ticket (TGT) forwarding, which makes Kerberos tickets available in order for SAS processes to authenticate to external resources. The Kerberos key distribution center (KDC) and the client configuration must allow forwarding.

Register the user principal in the Kerberos KDC. If the object spawner is running as *objspawn_user*, the principal in the KDC must be set to *objspawn_user@MY.KDC.DOMAIN*.

Include an entry for the principal, using the format *objspawn_user@MY.KDC.DOMAIN*, in the **objspawn.keytab** file. This file must be in the ObjectSpawner directory, must be owned by the service account user, and must have permissions set so that only the service account user can read the file.

For example, if the object spawner is running as sas and the Kerberos domain is MY.COMPANY.COM, you must register sas@MY.COMPANY.COM in the KDC and you must include an entry for sas@MY.COMPANY.COM in the **objspawn.keytab** file.

Creating the SASGRID Policy File

The SASGRID policy file specifies how SAS jobs use the resources with SAS Grid Manager for Hadoop. The policy file is an XML file that contains an entry for each type of processing that you want to perform on the grid (such as normal, priority, or night). Each type is identified as a GridApplicationType in the policy file. Within each application type entry, elements specify the details about how resources are allocated for those jobs. For example, an entry in the policy file for “normal” processing could specify the priority, memory, and number of cores used for jobs that should be run using normal processing.

Use the appType option on SAS jobs to specify the application type that a job should use. You can specify the appType in grid options. For example, including appType=normal in grid options specifies that jobs that use those grid options should use the parameters in the “normal” application type entry.

In addition, the SASGRID policy file contains <HostGroup> entries that enable you to specify which hosts in the Hadoop cluster are to be used for processing SAS grid jobs. If you want only a subset of hosts to be used for grid jobs, you can create a <HostGroup> entry that lists the available hosts, and then include that <HostGroup> entry in the application type entries. You can also create entries for groups of hosts that you want to use for certain application types. For example, you can create a <HostGroup> entry for hosts that you want to use only for priority processing. Although you can also specify hosts individually in application type entries, using

<HostGroup> enables you to specify groups that you can easily reuse in multiple entries.

If you do not create a SASGRID policy file, a job submitted using SAS Grid Manager for Hadoop can run on any host, and uses the maximum amount of memory and the minimum number of cores allowed by YARN (as defined by YARN properties in the **`yarn-site.xml`** file).

Use the policyFile grid option to specify the location of the policy file. You can specify a location on the file system of the grid client or a location in HDFS (prefix the path with **`hdfs://`**). If you do not specify a location using the policyFile option, SAS Grid Manager looks for a file named sasgrid-policy.xml in the directory that contains the cluster's configuration files. If the application does not find the file in that location, it searches for the path **`/tmp/SASGrid/sasgrid-policy.xml`** in HDFS. See "Supported Job Options – SAS Grid Manager for Hadoop" on page 311 for more information.

<GridPolicy defaultAppType="*name*"> is the top-level element in the file. Use the defaultAppType attribute on the GridPolicy element to specify which application type should be used if a submitted job does not include an `appType` value.

Within the <GridPolicy> element are a series of <GridApplicationType> entries. Each entry contains some or all of these elements:

<GridApplicationType name="*name*">
: Specifies the name for the application type. The name must be in quotation marks.

<jobname>
: Specifies the default job name if the application does not specify a name.

<priority>
: Specifies the priority for this job relative to other jobs. A higher number indicates a higher priority.

<nice>
: Specifies the execution priority change, based on Linux "nice" values. Valid values are integers in the range of -20 to 19.

<memory>
: The amount of physical memory (in megabytes) to reserve for the job.

<vcores>
: The number of virtual machine cores to allocate to the job.

<runlimit>
: The maximum time (in minutes) that the job is allowed to run.

<queue>
: Specifies the YARN scheduler queue that the job is submitted to. You create YARN queues when you set up YARN on your Hadoop cluster.

<hosts>
: Specifies the hosts and host groups that the job can run on. Hosts are specified using the <host> element, and host groups are specified using the <hostGroup> element.

A sample <GridApplicationType> entry looks like this:

```
<GridApplicationType name="normal">
            <jobname>Normal Job</jobname>
            <priority>2</priority>
            <memory>1024</memory>
            <vcores>1</vcores>
```

```
            <runlimit>120</runlimit>
            <queue>default</queue>
            <hosts>
                <hostGroup>development</hostGroup>
                <host>myhost4.mydomain.com</host>
            </hosts>
</GridApplicationType>
```

The <GridPolicy> element also includes <HostGroup> elements, which define host groups. Each <HostGroup> entry contains one or more <host> and <HostGroup> elements, which together specify which hosts are included in the group.

A sample <HostGroup> entry looks like this:

```
<hostGroup name="development">
        <host>myHost1.mydomain.com</host>
        <host>myHost2.mydomain.com</host>
</hostGroup>
```

A sample SASGRID policy file looks like this:

```
<?xml version ="1.0" encoding="UTF-8" standalone="yes"?>
<GridPolicy defaultAppType="normal">
        <GridApplicationType name="normal">
            <jobname>Normal Job</jobname>
            <priority>2</priority>
            <memory>1024</memory>
            <vcores>1</vcores>
            <runlimit>120</runlimit>
            <queue>default</queue>
            <hosts>
                <hostGroup>development</hostGroup>
                <host>myHost4.mydomain.com</host>
            </hosts>
        </GridApplicationType>

        <GridApplicationType name="priority">
            <jobname>High Priority</jobname>
            <priority>20</priority>
            <nice>-10</nice>
            <memory>1024</memory>
            <vcores>1</vcores>
            <queue>default</queue>
            <hosts>
                <hostGroup>test</hostGroup>
            </hosts>
        </GridApplicationType>

        <HostGroup name="development">
            <host>myHost1.mydomain.com</host>
            <host>myHost2.mydomain.com</host>
        </HostGroup>

        <HostGroup name="test">
            <host>myHost3.mydomain.com</host>
            <hostGroup>development</host>
        </HostGroup>

</GridPolicy>
```

Sending Jobs to SAS Grid Manager for Hadoop

After you have created the SASGRID policy file, you can use application types that you defined to identify how SAS jobs are to be processed by the grid. Use any of the following methods to specify an application type to be associated with any jobs submitted to the SAS grid:

Grid server definition
: Specify **AppType=*application_type*** in the **Grid Options** field of the YARN grid server definition (specify **YARN** in the **Grid Provider** field). This value is used if an AppType value is not otherwise specified when a job is submitted to the grid, so you might want to specify the application types used for normal processing. Specify *application_type* in the `jobOpts=` parameter of the `grdsvc_enable` function.

grdsvc_enable function
: Specify application_type in the jobOpts= parameter of the grdsvc_enable function.

Grid options set
: Grid options sets enable you to create sets of SAS options, required resources, and grid options. Each options set can then be mapped to a specified combination of a SAS application and a user or user group. When you create grid options sets for Grid Manager for Hadoop, go to the New Grid Options Set window and specify **AppType=*application_type*** in the **Grid Options** field. Specifying this value enables you to control the type of processing for jobs on the grid. See Chapter 14, “Working With Grid Options Sets,” on page 247 for more information.

Note: If you are using SAS Grid Manager for Hadoop and specify options in a grid options set that are specific to a Platform Suite for SAS grid, those options are ignored.

SASGSUB statements
: Specify `AppType=`*`application_type`* as an option for the -GRIDJOBOPTS argument in a SASGSUB statement. You can also specify the application type in a grid options set and specify the grid options set in the -GRIDOPTSET argument. See “SASGSUB Command: Grid Job Arguments” on page 300.

If you are using SAS Grid Manager for Hadoop and the SAS Grid Manager Client Utility, you can restart grid jobs by using checkpoints and labels.See “Using SAS Checkpoint and Label Restart” on page 255 for more information.

Troubleshooting SAS Grid Manager for Hadoop

Debugging Grid Jobs

When a grid-enabled SIGNON is used or a job is submitted by SASGSUB, the sasgrid script is used to start a SAS session on the grid node. To aid in debugging, the output from the script (but not the SAS log and SAS output) is written to a unique log file. The file is located in the **`GridServer/Logs`** directory of the machine where the job runs. If the job is successful, the log file is deleted when the job finishes running. If the job is unsuccessful, the log file is saved.

The log file is named using the format SASGrid.*userID.hostname.process_ID.YYYMMDDhhmmss*.log

Debugging YARN

Log Sources

To debug SAS Grid Manager for Hadoop jobs in YARN, you can use the Hadoop Resource Manager application, and you can use information from these sources:

- SAS
- SAS Yarn AppMaster
- YARN server logs

YARN commands, and log entries. There are several types of logs that provide information about SAS Grid operation, and their locations differ depending on which grid provider you are using.

SAS Logging

Using the SAS logging facility, set the `app.tk.tkegrid.yarn` logger value to debug or trace so that logging messages are sent to the appender.

You can also use the SASGrid.*.log file, which is located at ***`SASconfig`***`/Levn/`***`SASAppServerContext`***`/GridServer/Logs`.

SAS YARN AppMaster Logs

You can access the YARN AppMaster logs by using the Hadoop Resource Manager application or the command `yarn logs -applicationID <applicationID>`.

YARN Server Logs

The YARN Resource Manager log is contained in the file **`*-resourcemanager-<host>.log`**. The Node Manager log is contained in the file **`*-nodemanager-<host>.log`**. The location of these logs differs depending on the Hadoop distribution that you are using:

Hortonworks
: **`/var/log/hadoop-yarn`**

Cloudera
: **`/var/log/hadoop-yarn`**

MapR
: **`/opt/mapr/hadoop/hadoop-<ver>/logs`**

Scheduling with SAS Grid Manager for Hadoop

SAS Grid Manager for Hadoop uses Oozie for scheduling SAS workflows. To implement scheduling using Oozie, you define an Oozie scheduling server in SAS metadata and specify that server when deploying SAS DATA step jobs for scheduling. Only SAS DATA Step jobs are supported for scheduling. When a SAS DATA Step job is sent to the server, it generates these two XML files:

workflow file
: contains a list of tasks that must be performed for the scheduled job

coordinator file
: contains the triggers that control when the job is to be run

Because of the functions supported by Oozie, the Oozie scheduling server provides support only for simple schedules, allowing only a single time event as a trigger, supporting only job events as flow dependencies, and only allowing AND conditions in a process flow diagram.

See *Scheduling in SAS* for complete information on scheduling with Grid Manager for Hadoop using an Oozie scheduling server.

PART 5

Common SAS Grid Manager Features

12

Installing and Configuring Common Components

Overview

Some aspects of configuring and using SAS Grid Manager are the same regardless of whether you are using SAS Workload Orchestrator, Platform Suite for SAS, or SAS Grid Manager for Hadoop as your grid provider. These parts of the SAS grid infrastructure are used in both environments and follow the same procedure for installation and configuration:

- logical grid server definitions
- SAS Grid Manager Client Utility

Likewise, the methods that end users of SAS applications use to submit jobs to SAS Grid Manager do not differ between grid providers. From the user's perspective, they do not know (and do not need to know) whether the job that they submit is being processed on a Platform Suite for SAS grid or on a SAS Grid Manager for Hadoop grid.

Common Metadata Definitions

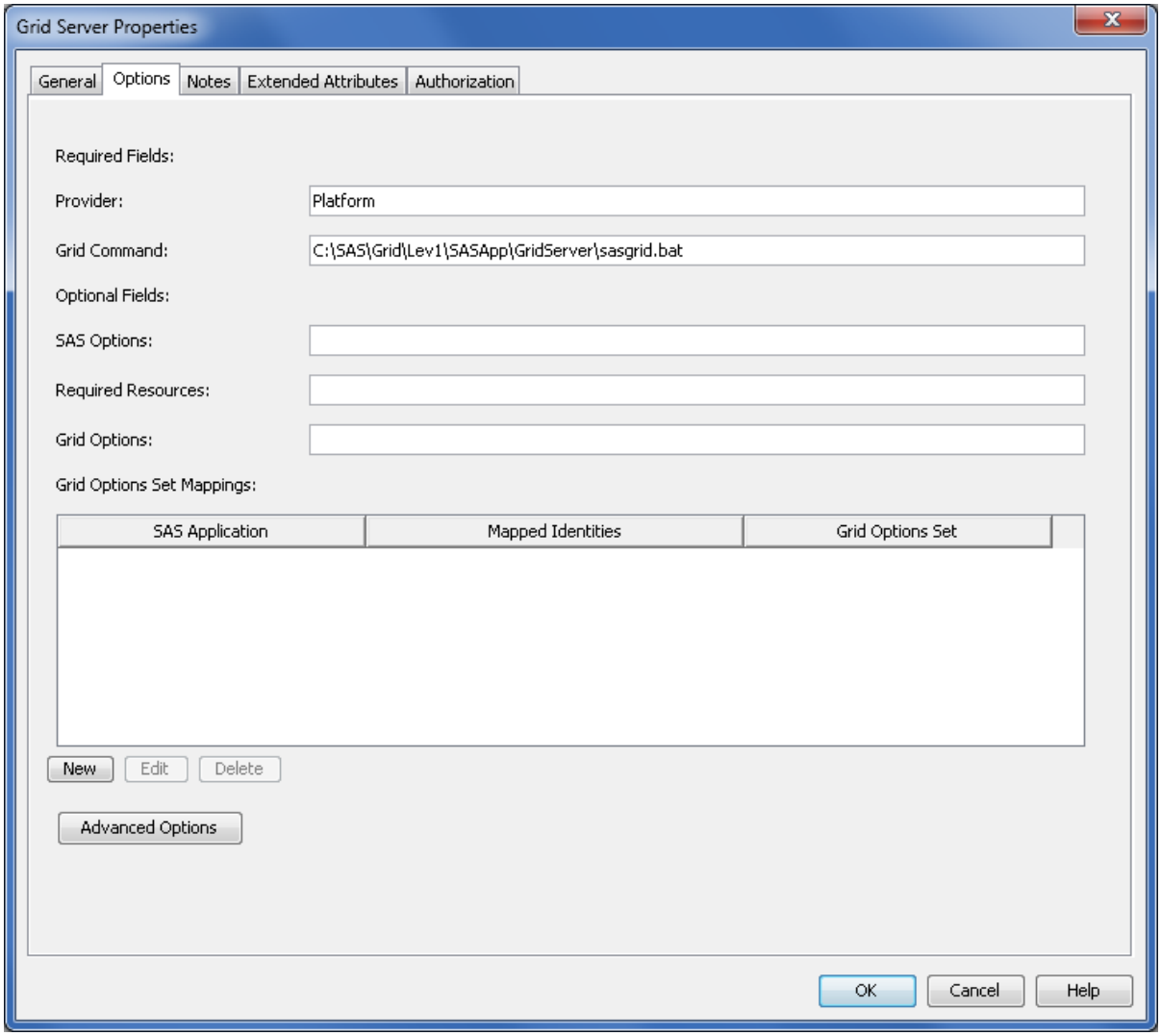
Modifying SAS Logical Grid Server Definitions

The initial configuration of the logical grid servers is performed by the SAS Deployment Wizard. However, a SAS grid administrator might need to modify the existing grid metadata or add new grid metadata definitions.

A SAS administrator performs these steps to specify or modify the required and optional properties as metadata for the SAS Grid Server:

1. In SAS Management Console, open the metadata repository that contains the metadata for the Logical Grid Server.
2. In the navigation tree, select **Server Manager**.
3. Expand the folders under **Server Manager** until you see the metadata objects for the SAS Application Server, such as `SASApp`, and its Logical Grid Server component.
4. Expand the Logical Grid Server component so that you see the metadata object for the Grid Server.
5. Right-click the metadata object for the Grid Server, and select **Properties**.
6. In the Grid Server Properties window, click the **Options** tab.

Figure 12.1 Grid Server Properties



7 The fields on the **Options** tab are:

Provider
the grid middleware provider. This value is used to communicate with the grid control server.

Grid Command
the script, application, or service that Platform Suite for SAS uses to start server sessions on the grid nodes. Any SAS options that are included in this command are passed to the grid jobs. You can specify any additional SAS options in the **SAS Options** field.

This value is the path to the sasgrid.cmd file (Windows) or sasgrid script file (UNIX). Because this same command is used to start the servers on all grid nodes, the path to the directory on each grid node must be the same. For example: `C:\SAS\Grid\Lev1\SASApp\GridServer\sasgrid`.

SAS Options
specifies the SAS options that are applied by default to jobs that are sent to the grid. Separate multiple options with a space. Any SAS options that are specified for a specific application mapping override this value.

Required Resources
specifies the name of the resources that are applied by default to jobs that are sent to the grid. Any resources that are specified for a specific application mapping are added to this value. The application server name is displayed in this field by default, but it can be removed if needed.

Grid Options
specifies the grid command options that are applied by default to jobs that are sent to the grid. Examples include the job priority, the job queue, or the user group that is associated with the job. Any grid options that are specified by a grid options set and mapped to a specific application override this value. Job options are specified as name/value pairs in this format. Multiple pairs are separated by semicolons:

```
option-1=value-1;option-2="value-2 with spaces";
... option-n='value-n with spaces';
```

Here is an example of additional options that specify that all jobs that use this logical grid server go to the priority queue in the project “payroll”:

```
queue=priority; project='payroll'
```

For a complete list of job options for SAS Workload Orchestrator, see “Supported Job Options – SAS Workload Orchestrator” on page 305.

For a complete list of job options for Platform Suite for SAS, see Appendix 2, “Supported Job Options - SAS Grid Manager for Platform,” on page 307.

For a complete list of job options for SAS Grid Manager for Hadoop, see “Supported Job Options – SAS Grid Manager for Hadoop” on page 311.

Grid Options Set Mappings
lists the grid server’s mappings of grid options sets to applications for a specified user or group. Each grid options set associates a set of SAS options, required resources, and grid options to a SAS application and user identities that use the grid. To change or remove an entry, click on the entry and select **Edit** or **Delete** from the pop-up menu. Click **New** to create a new application mapping. See Chapter 14, “Working With Grid Options Sets,” on page 247 for more information about grid options sets.

Advanced Options
enables you to specify the workload and module name. The workload specifies the optional resources that can be processed on the node. The module name specifies the shared library name or Java class name of the grid middleware provider's support plug-in.

8 After you complete the field entries, click **OK** to save the changes and close the Grid Server Properties window.

9 In the display area (right-hand side) on SAS Management Console, right-click the **Connection** object for the Grid Server, and then select **Properties**.

10 In the Properties window for the Grid Server Connection, click the **Options** tab. Here are the fields on this tab:

Authentication Domain
the authentication domain that is used for connections to the server. If you are using SAS Workload Orchestrator, see “Configuring SAS Workload Orchestrator Authentication” on page 29 for information about selecting this value. If you are using SAS Grid Manager for Platform or SAS Grid Manager for Hadoop, set this value to **<none>**.

Grid Server Address
the host name or network address of the grid control server.

Grid Server Port
the port that is used to connect to the grid control server. For SAS Workload Orchestrator, this value should be set to the port on which SAS Workload Orchestrator is listening. See "Configuring SAS Workload Orchestrator General Options" on page 41. This value should always be set to 0 (zero) for Platform LSF.

Installing and Configuring SAS Grid Manager Client Utility

Installation Overview

The SAS Grid Manager Client Utility enables users to submit SAS programs to a grid for processing without requiring SAS to be installed on the machine that is performing the submission. If you are using SAS Grid Manager for Platform, Platform Suite for SAS must be installed on any machine on which the SAS Grid Manager Client Utility runs.

The SAS Grid Manager Client Utility is automatically installed and configured using the SAS Deployment Wizard if the utility is in the plan file.

Installation Prerequisites

The configuration for the SAS Grid Manager Client Utility assumes that all of the following actions have been performed:

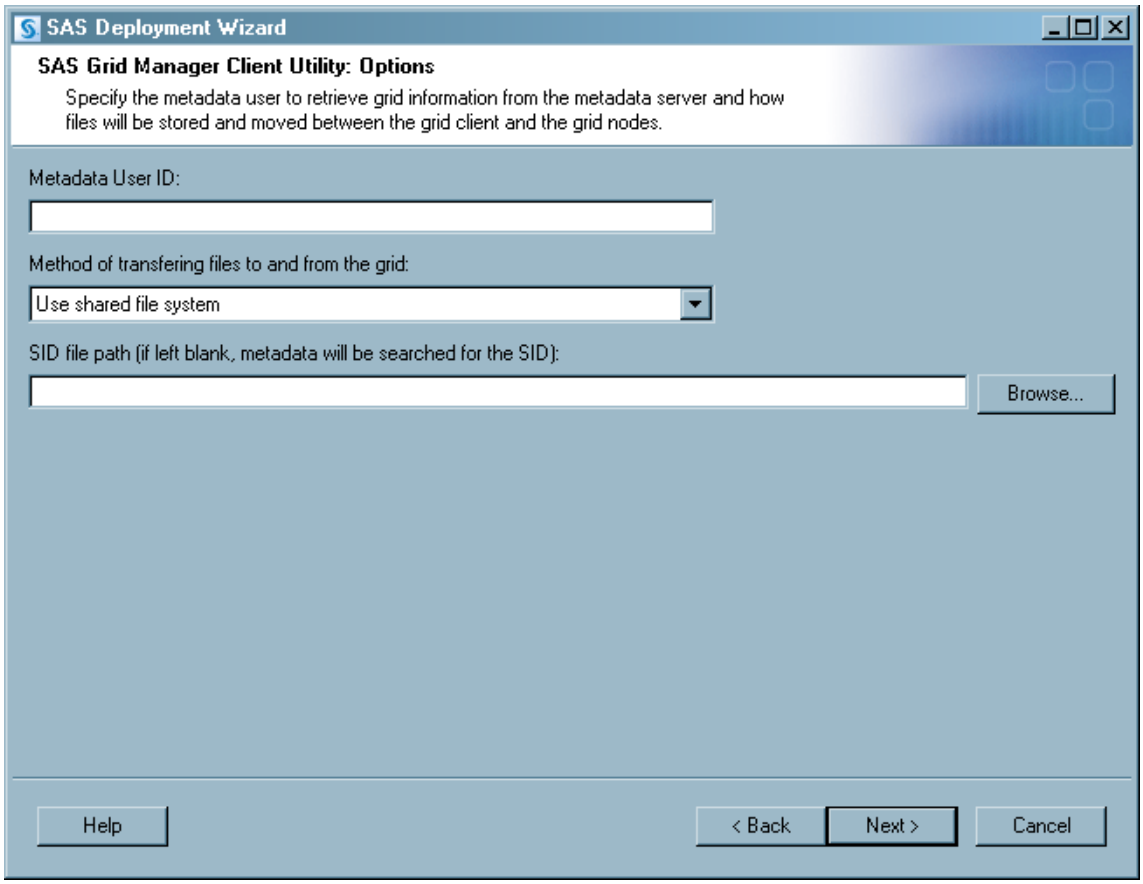
- The grid control server has already been installed. The configuration must retrieve the logical grid server definition from metadata.
- The user name under which jobs are submitted is defined in metadata. If not, jobs submitted to the grid fail.

Configuring the SAS Grid Manager Client Utility

The amount of user input that is required during the installation and configuration process depends on whether you choose an Express, Typical, or Custom install. For information about running the SAS Deployment Wizard, see *SAS Deployment Wizard User's Guide*.

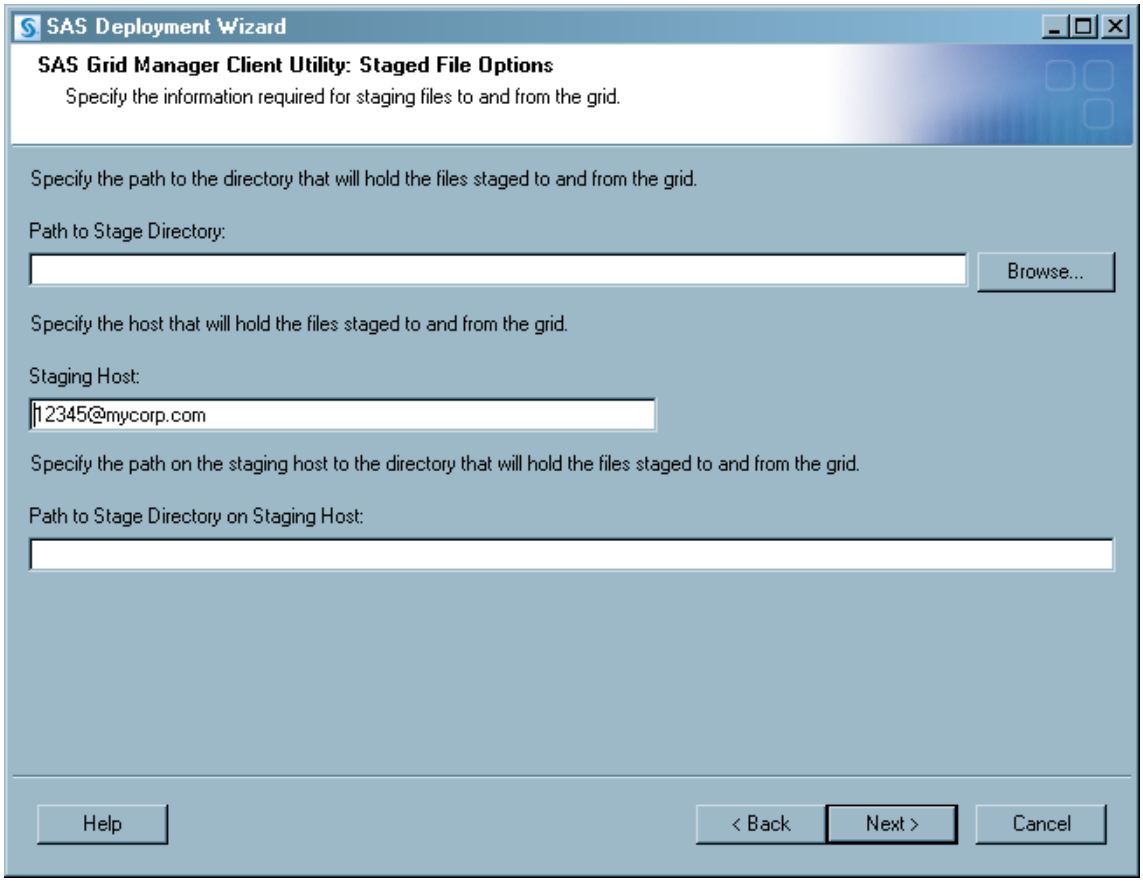
1 The SAS Grid Manager Client Utility: Options page enables you to specify the user credentials that are used to connect to the SAS Metadata Server, the method for transferring files to and from the grid (either through a shared file system or remote copy), and the path to a SAS license file that contains a SAS Grid Manager license (only shown during a custom installation). By default, the metadata is searched for the SAS license file.

Figure 12.2 Grid Manager Client Utility: Options Page



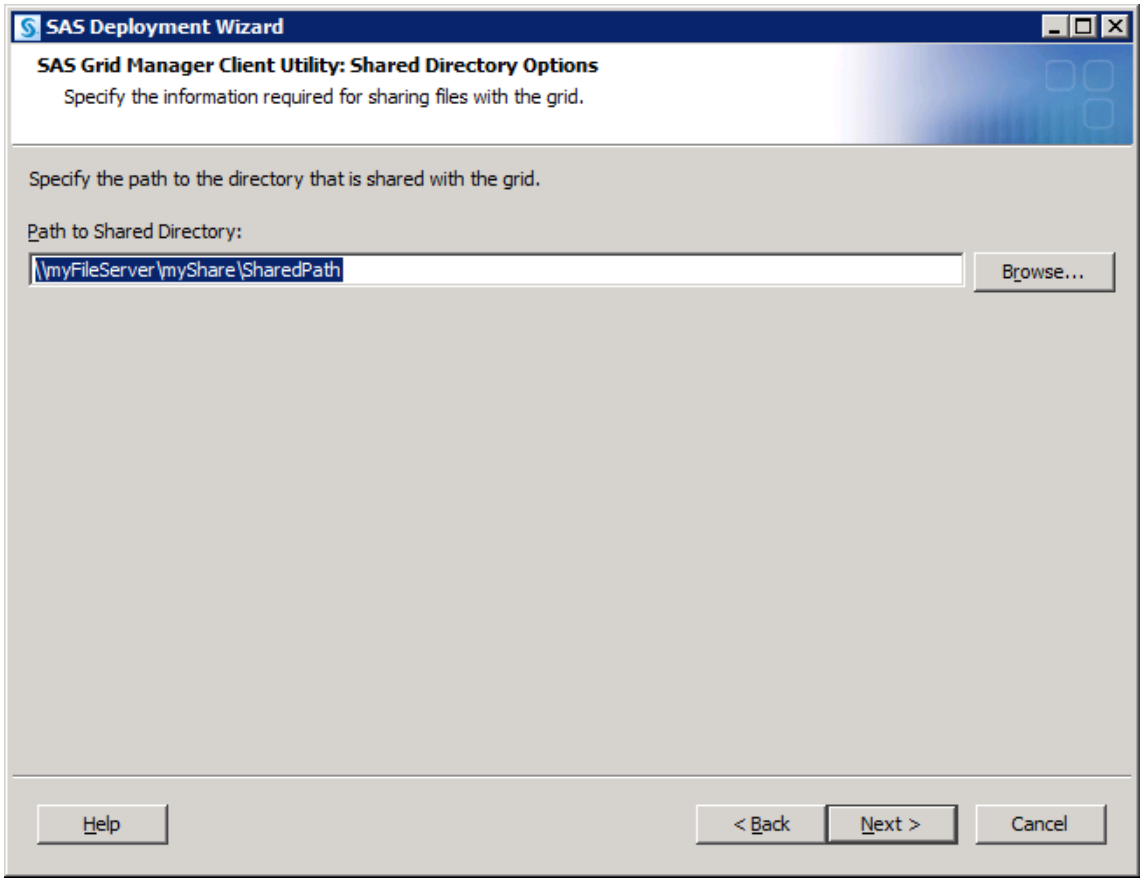
2 If you choose to use remote copy (also known as staging) to transfer files to and from the grid, the SAS Grid Manager Client Utility: Staged File Options page is displayed. This page enables you to specify the path to the directory used to stage files moving into and out of the grid. If you are performing a custom installation, you can also specify the staging host and the path to the staging directory as seen by the staging host.

Figure 12.3 *SAS Grid Manager Client Utility: Staged File Options Page*



If you choose to use a shared directory to copy files to and from the grid, the SAS Grid Manager: Shared Directory Options page appears. This page enables you to specify the grid-shared directory on the grid control server.

Figure 12.4 SAS Grid Manager Client Utility: Shared Directory Options Page



Using the SASGSUB Configuration File

Most of the options that are used by the SAS Grid Manager Client Utility are contained in the sasgsub.cfg file. This file is automatically created by the SAS Deployment Wizard. These options specify the information that the SAS Grid Manager Client Utility uses every time it runs. The sasgsub.cfg file is located in the Applications/SASGridManagerClientUtility/*version* directory of the configuration directory. The following information from the SAS Deployment Wizard is collected in the sasgsub.cfg file:

- information to connect to the SAS Metadata Server (SAS Metadata Server name, port, user ID, and password). By default, the metadata password value is set to _PROMPT_, and the user is prompted for a password.
- the path that is used to store files used by the grid. If you are using a shared file system, then this is the path to the shared file system. If you are staging files, then this is the location where grid clients store files that are retrieved by the grid.
- the name of the SAS Application Server that contains the logical grid server definition.

13

Enabling SAS Applications to Run on a Grid

Overview of Grid Enabling

After you have configured your grid, you can configure your SAS applications and programs to take advantage of the grid capabilities. Some SAS applications require you to change only an option to take advantage of the grid; other applications require more extensive changes.

Submitting SAS Programs to a Grid Interactively

Overview

You can use SAS Display Manager as a client to submit SAS programs to the grid for execution. The results of the execution are returned to the local workstation. When you submit a SAS program from a SAS Display Manager client to execute on a grid, the program runs on a grid machine in a separate session with its own unique work library. The SAS log and output of the grid execution are returned to the local workstation. You might need to perform additional actions in order to view data from the SAS Display Manager session that was created or modified by the program that ran on the grid. For example, modifications might be required in order to use the Explorer to browse SAS libraries that are modified by grid execution.

Starting with SAS 9.4M1, you can also use the SAS Grid Manager Client Utility to start an interactive session on the grid or watch the output from a batch session. An

interactive session enables you to perform functions in interactive mode or batch monitor mode:

Interactive mode

- Run SAS in line mode.
- Run SAS in Display Manager mode.
- Run and interact with a command (watch the output and provide input).

Batch monitor mode

- Run SAS in batch mode and watch the output of the session.
- Run a command in batch mode and watch the output.

Using SAS line mode or SAS Display Manager mode requires that the grid nodes are UNIX machines. Also, SAS Display Manager mode requires that you have X Server on the client machine.

If you terminate the SAS Grid Manager Client Utility session while performing any interactive mode function, the remote job is also terminated. Terminating the session while running a batch monitor mode function does not terminate the batch job.

Submitting Jobs from the Program Editor to the Grid

The first step in integrating SAS processes with the grid is to get your SAS programs running on the grid.

In order to submit a SAS program to the grid, you must add a set of grid statements to the program. For programs submitted through the SAS Program Editor, you can save the statements to an external file and then specify a key definition that issues the statements. Submit the contents of the SAS Program Editor window to the grid, rather than to the local workstation.

Some of the examples in this topic use SAS/CONNECT statements (such as signon, rsubmit, and signoff). For detailed information about these statements, see *SAS/CONNECT User's Guide*.

Note: This procedure does not work if the Explorer window is open in your SAS session.

To add grid statements to a program and submit the program to the grid, follow these steps:

1. Save these statements to an external file, referred to as *grid-statement-file* (for example, c:\gpre.sas):

```
%global count;
%macro gencount;
%if %bquote(&count) eq %then %do; %let count=1;%end;%else %let
count=%eval(&count+1);
%mend;
%gencount;
options
metaserver='metadata-server-address';
options metaport=metadata-server-port;
options metauser=username;
options metapass="password";
```

```
%let rc=%sysfunc(grdsvc_enable(grid&count, server=SASApp));
 signon grid&count;
```

metadata-server-address is the machine name of the SAS Metadata Server, and *metadata-server-port* is the port used to communicate with the metadata server.

2 Open the Keys window and specify the following for an available key (for example, F12):

```
gsubmit "%include
'grid-statement_file';";
rsubmit grid&count wait=no persist=no;
```

grid-statement-file is the path and filename of the file (for example, c:\gpre.sas) containing the grid statements.

3 Type or include a SAS program in the Program Editor window, and then press the key assigned to the grid statements. The program is automatically submitted to the grid for processing. Your local machine is busy only until the program is submitted to the grid.

Using the same key to submit multiple jobs causes multiple jobs to be executed in parallel on the grid. Anything specified in the WORK library is discarded when the grid session ends and is not available to later steps in the code.

Viewing LOG and OUTPUT Lines from Grid Jobs

The example in "Submitting Jobs from the Program Editor to the Grid" on page 219 uses asynchronous rsubmits. This causes the results of the execution to be returned to the local log and output windows only after the entire program finishes execution on the grid. To cause the log and output lines to be displayed while the program is executing, delete the `options noconnectwait;` line in the program.

The rsubmit executes synchronously, and the returned log and output lines are displayed while the job is executing. This also results in the Client SAS session being busy until the entire grid job has completed. You cannot submit more code until the job completes.

Using the SAS Explorer Window to Browse Libraries

The Client SAS session and the grid SAS session are two separate instances of SAS. Any code or products needed to access data must be submitted and available on both the client machine as well as the grid nodes. Use the following steps to browse libraries from the SAS Explorer Window that are accessed and modified by jobs executing on the grid:

1 Define all of your SAS libraries within SAS metadata under your server context (for example, under SASApp).

2 Ensure that the following option is in the SAS invocation in the sasgrid script file used to start SAS on the grid nodes. This option should have been added by the SAS Deployment Wizard.

```
metaautoresources SASApp
```

SASApp is the name of your application server context.

3 Include this option on the Client SAS session invocation on the workstation.

```
metaautoresources SASApp
```

SASApp is the name of your application server context.

Note: If you are accessing data through any SAS/ACCESS product, you must license the SAS/ACCESS products on the SAS Client machine in order to be able to browse those libraries from the SAS Explorer. The SAS/ACCESS products must also be licensed on the grid nodes in order to enable the job to access data during execution.

Each SAS session executing on the grid is a unique session with a unique WORK library. In order to view the work libraries that are created on each of the grid nodes, you must add the following line after the signon statement in the code provided in "Submitting Jobs from the Program Editor to the Grid" on page 219:

```
libname workgrid slibref=work server=grid&count;
```

`grid&count` is the label used as the remote session ID in the signon statement.

Running Interactive SAS Sessions Using the SAS Grid Manager Client Utility

To start an interactive SAS session on a UNIX grid using the SAS Grid Manager Client Utility and SAS Display Manager, issue the following command from an operating system command line:

<*path*/>SASGSUB -GRIDRUNSASDMS <*hostName*>:*display_number*<.*screen_number*>

To start an interactive SAS session in line mode on a UNIX grid using the SAS Grid Manager Client Utility, issue the following command from an operating system command line:

<*path*/>SASGSUB -GRIDRUNSASLM

The *path* option specifies the path for the SASGSUB program. By default, the location is <*configuration_directory*>/Applications/SASGridManagerClientUtility/*version*.

The -GRIDRUNSASDMS or -GRIDRUNSASLM option starts the interactive SAS session on the grid. Once the session has started, you can submit SAS code or programs that will run on the machine on the grid. There are other arguments, some required and some optional, that you can include in the SASGSUB statement. These arguments are passed to the grid when the SAS session is started and include parameters such as the metadata connection information, grid application server name, and workload resource names. See "SASGSUB Command: Starting SAS in Interactive Mode" on page 285 for complete syntax information.

Running Interactive Commands Using the SAS Grid Manager Client Utility

To interactively run a command on a grid using the SAS Grid Manager Client Utility, issue the following command from a SAS command line:

<*path*/>SASGSUB -GRIDRUNCMDINT *command*

Standard input, standard output, and standard error are directed to your command prompt window.

Submitting Batch SAS Jobs to the Grid

Overview

The SAS Grid Manager Client Utility enables you to run SAS jobs on the grid in batch. You can also use the utility to check job status, end a job, and retrieve job output. Most of the options that are used by the SAS Grid Manager Client Utility are contained in the sasgsub.cfg file. This file is automatically created by the SAS Deployment Wizard. These options specify the information that the SAS Grid Manager Client Utility uses every time it runs.

The SAS Grid Manager Client Utility and Platform LSF must be installed on any machine on which the SAS Grid Manager Client Utility runs.

Grid Manager Client Utility File Handling

This is how files are handled by the SAS Grid Manager Client Utility when processing a job on the grid in batch mode:

1. SASGSUB creates a job directory in the GRIDWORK directory under the directory of the user who is submitting the job. For example, if GRIDWORK is `/grid/share` and the submitting user is `sasuser1`, then a job directory is created in `/grid/share/sasuser1` for the files.
2. SASGSUB copies the SAS program and any files specified by GRIDFILESIN into the new directory.
3. SASGSUB submits a job to the grid that includes information about the location of the job directory. It uses either GRIDWORK or GRIDWORKREM to specify the location of the job information to the grid. If you are staging files, SASGSUB also passes the stage file command specified by the GRIDSTAGECMD option to the grid.

4 If the grid job is using staging when the job starts, the grid copies the files in the job directory under GRIDWORK to a temporary job directory. The temporary directory is in the grid's shared directory location that is specified during the SAS Deployment Wizard installation process.

5 The grid runs the SAS program from the job directory and places the LOG and LST file back into the same job directory. For a shared file system, this directory is the one specified by the GRIDWORK option. This is also the directory that SASGSUB copied files into. If you are staging files, this directory is the job directory that is in the grid shared directory.

6 If you are staging files, after the job is complete, the files in the job directory in the grid shared location are copied to the job directory that is specified by the GRIDWORK option.

7 At this point in processing, the job directory in GRIDWORK contains all of the files that are required and produced by SAS batch processing. You can then retrieve the files using the GRIDGETRESULTS command.

Submitting Jobs in Batch Using the SAS Grid Manager Client Utility

To submit a SAS job in batch mode to a grid using the SAS Grid Manager Client Utility, issue the following command from an operating system command line:

<*path*/>SASGSUB -GRIDSUBMITPGM *sas-program-file*

The *path* option specifies the path for the SASGSUB program. By default, the location is *configuration_directory*/Applications/SASGridManagerClientUtility/*version*.

The -GRIDSUBMITPGM option specifies the name and path of the SAS program that you want to submit to the grid.

In addition, you can specify other options that are passed to the grid or used when processing the job, including workload resource names. For a complete list of options, see “SASGSUB Command: Submitting a SAS Program in Batch Mode” on page 282.

Specifying the -GRIDWATCHOUTPUT argument displays the standard output and standard error of the submitted batch job on your machine.

Running Commands in Batch Using the SAS Grid Manager Client Utility

To submit a command to a grid in batch mode using the SAS Grid Manager Client Utility, issue the following command from an operating system command line:

<*path*/>SASGSUB -GRIDRUNCMD*command*

Viewing Job Status Using the SAS Grid Manager Client Utility

After you submit a job to the grid, you might want to check the status of the job. To check the status of a job, issue the following command from a command line:

<*path*/>SASGSUB -GRIDGETSTATUS [*job-ID* | ALL]

`-GRIDGETSTATUS` specifies the ID of the job that you want to check, or ALL to check the status of all jobs submitted by your user ID. For a complete list of options, see "SASGSUB Command: Viewing Job Status" on page 288.

Here is an example of the output produced by the SASGSUB -GRIDGETSTATUS command.

Output 13.1 *Output Produced by SASGSUB -GRIDGETSTATUS Command*

```
Current Job Information
  Job 1917 (testPgm) is Finished:  Submitted: 08Dec2008:10:28:57, Started: 08Dec2008:10:28:57 on
Host host1, Ended: 08Dec2008:10:28:57
  Job 1918 (testPgm) is Finished:  Submitted: 08Dec2008:10:28:57, Started: 08Dec2008:10:28:57 on
Host host1, Ended: 08Dec2008:10:28:57
  Job 1919 (testPgm) is Finished:  Submitted: 08Dec2008:10:28:57, Started: 08Dec2008:10:28:57 on
Host host1, Ended: 08Dec2008:10:28:57
  Job information in directory U:\pp\GridSub\GridWork\user1\SASGSUB-2008-11-24_13.17.17.327_testPgm
is invalid.
  Job 1925 (testPgm) is Submitted: Submitted: 08Dec2008:10:28:57
```

Ending Jobs Using the SAS Grid Manager Client Utility

If a job that has been submitted to the grid is causing problems or otherwise needs to be terminated, use the SAS Grid Manager Client Utility to end the job. Issue the following command from a command line:

<*path*/>SASGSUB -GRIDKILLJOB [*job-ID* | ALL]

`-GRIDKILLJOB` specifies the ID of the job that you want to end, or ALL to end all jobs submitted by your user ID. For a complete list of options, see "SASGSUB Command: Ending a Job" on page 288.

Retrieving Job Output Using the SAS Grid Manager Client Utility

After a submitted job is complete, use the SAS Grid Manager Client Utility to retrieve the output produced by the job. Issue the following command from a command line:

<*path*/>SASGSUB -GRIDGETRESULTS [*job-ID* | ALL] -GRIDGETRESULTSDIR

-GRIDGETRESULTS specifies the ID of the job whose results you want to retrieve, or you can specify ALL to retrieve the results from all jobs submitted by your user ID.

-GRIDRESULTSDIR specifies the directory in which the jobs results should be moved. When the results are retrieved, they are removed from the GRIDWORK directory, which keeps this directory from filling up with completed jobs. If you do not specify this parameter, the results are copied to a job subdirectory in the current directory.

A file named job.info is created along with the job output. This file contains information about the execution of the job, including the submit time, start time, end time, the machine on which the job ran, the job ID, and the return code from the SAS program.

Here is an example of the output produced by the SASGSUB -GRIDGETRESULTS command.

Output 13.2 *Output Produced by SASGSUB -GRIDGETRESULTS Command*

```
Current Job Information
  Job 1917 (testPgm) is Finished:  Submitted: 08Dec2008:10:53:33, Started: 08Dec2008:10:53:33 on
Host host1, Ended: 08Dec2008:10:53:33
    Moved job information to .\SASGSUB-2008-11-21_21.52.57.130_testPgm

  Job 1918 (testPgm) is Finished:  Submitted: 08Dec2008:10:53:33, Started: 08Dec2008:10:53:33 on
Host host1, Ended: 08Dec2008:10:53:33
    Moved job information to .\SASGSUB-2008-11-24_13.13.39.167_testPgm

  Job 1919 (testPgm) is Finished:  Submitted: 08Dec2008:10:53:34, Started: 08Dec2008:10:53:34 on
Host host1, Ended: 08Dec2008:10:53:34
    Moved job information to .\SASGSUB-2008-11-24_13.16.06.060_testPgm

  Job 1925 (testPgm) is Submitted: Submitted: 08Dec2008:10:53:34
```

Retrieving a SAS Grid Manager Client Utility Log

After a submitted job is complete, you can find the SAS program log file for the job in this location: **GRIDWORK/*user id*/SASGSUB-*YYYY-MM-DD_HH:MM_SS_mmm_job_name*/*program_name*.log**

The SAS Grid Manager Client Utility uses the standard SAS logging facility. Output from the SAS Grid Manager Client Utility is directed to the console unless you use the SAS logging facility to create a log.See the -LOGCONFIGLOC option in

"SASGSUB Command: Submitting a SAS Program in Batch Mode" on page 282 for a list of the supported logging keys.

Monitoring Batch Processing Using the SAS Grid Manager Client Utility

When you use SASGSUB to submit a program or a command to the grid in batch mode, you can use the -GRIDWATCHOUTPUT argument to interactively monitor the processing on the grid. The option specifies that the output of what was submitted by the SASGSUB command is displayed on your machine. If you use this argument when submitting a SAS program using -GRIDSUBMITPGM, the SAS log and output are displayed. If you use the argument when submitting a command using -GRIDRUNCMD, the command's standard output and standard error are displayed. While the output is being displayed, entering the command prompt does not affect the processing on the grid.

If you terminate the SASGSUB session while in interactive monitoring mode, the batch job continues to run and does not terminate.

Using a Grid without a Shared Directory

If your grid configuration does not permit a directory structure to be shared between the grid client machines and the grid nodes, you can specify that the grid job move files into the grid before processing and move files out of the grid when the job is complete. The file movement (called file staging) is performed by the grid job using a remote copy program such as rcp, scp, or lsrcp. When using file staging, files are moved into and out of the grid using the GRIDWORK directory. The SAS Grid Manager Client Utility passes information to the grid that indicates which files need to be sent to the grid and where the files are located. After the grid processes the job, the results are copied back to the GRIDWORK directory. If the user is offline, the results are held in the shared file system until they are retrieved.

During the installation process, the SAS Deployment Wizard enables you to specify whether you will use a shared directory or if you will be staging files. If you specify that you will be staging files, you must also specify the staging command that you want to use to move the files (rcp, lsrcp, scp, pscp, or smbclient). You can also specify the host that you will use to stage files to and from the grid, if you are not using the current host.

To submit jobs to a grid without a shared file system, follow these steps:

1. Use the GRIDSTAGECMD parameter on the SASGSUB command to specify the transfer method to use for moving the files from the staging directory to the grid.
2. If the machine that stages the files is not the current host, use the GRIDSTAGEHOST parameter on the SASGSUB command to specify the host that is used to stage the files. For example, use this parameter if you are using a laptop to submit jobs to the grid and then disconnecting or shutting down the laptop before the jobs are completed or submitted. The laptop must have a GRIDWORK directory on a file server that is always available to the grid. Use the GRIDSTAGEHOST command to specify the file server host name.

Scheduling Jobs on a Grid

Using the scheduling capabilities, you can specify that jobs are submitted to the grid when a certain time has been reached or after a specified file or job event has occurred (such as a specified file being created).

To schedule a job to run on a grid, follow these steps:

1. Deploy the job for scheduling.

   Some SAS applications, such as SAS Data Integration Studio, include an option to deploy jobs for scheduling. If you want to schedule an existing SAS job, use the **Deploy SAS DATA Step Program** option in the Schedule Manager plug-in of SAS Management Console.

2. Use the Schedule Manager plug-in in SAS Management Console to add the job to a flow.

   A flow contains one or more deployed jobs as well as the schedule information and time, file, or job events that determine when the job runs.

3. Assign the flow to a scheduling server and submit the flow for scheduling.

   You must assign the flow to a Platform Process Manager scheduling server in order for the scheduled job to run on the grid.

For detailed information about scheduling, see *Scheduling in SAS*.

Comparing Grid Submission Methods

You can use the SAS Grid Manager Client Utility, the Schedule Manager plug-in to SAS Management Console, and SAS language statements to submit jobs to the grid. The following table compares the methods.

Table 13.1 *Comparison of Grid Submission Methods*

Feature	SAS Grid Manager Client Utility	Schedule Manager Plug-in	SAS Language Statements
Interface	Command line	SAS Management Console interface	SAS language syntax

Feature	SAS Grid Manager Client Utility	Schedule Manager Plug-in	SAS Language Statements
Duration of client connection	Duration of the submission	Duration of the submission	Duration of the execution
Minimum client installation requirements	SAS Grid Manager Client Utility and Platform LSF	SAS Management Console	Base SAS, SAS/CONNECT , Platform LSF
Support for checkpoint restart	Yes	Yes	No
Support for SAS options, grid options, and policies	Yes	Yes	Yes
Support for event-triggered workflow execution	No	Yes	No

Enabling Distributed Parallel Execution of SAS Jobs

Some SAS programs contain multiple independent subtasks that can be distributed across the grid and executed in parallel. This approach enables the application to run faster. To enable a SAS program to use distributed parallel processing, add RSUBMIT and ENDRSUBMIT statements around each subtask and add the GRDSVC_ENABLE function call. SAS Grid Manager or SAS Grid Manager for Platform automatically assigns each identified subtask to a grid node.

You can use the SAS Code Analyzer to automatically create a grid-enabled SAS job. To use the SAS Code Analyzer, add PROC SCAPROC statements to your SAS program, specifying the GRID parameter. When you run the program with the PROC SCAPROC statements, the grid-enabled job is saved to a file. You can then run the saved SAS job on the grid, and SAS Grid Manager or SAS Grid Manager for Platform automatically assigns the identified subtasks to a grid node.

This is an example of the syntax for the SAS Code Analyzer:

```
proc scaproc;
     record '1.txt' grid '1.grid':
run;
remainder of SAS program...
```

For complete information and syntax for the PROC SCAPROC statement, see *Base SAS Procedures Guide*.

This is an example of the syntax used for enabling distributed parallel processing:

```
% let rc=%sysfunc(grdsvc_enable(_all_,server=SASApp));
options autosignon;
rsubmit task1 wait=no;
    /* code for parallel task #1  */
endrsubmit;
rsubmit task2 wait=no;
    /* code for parallel task #2 */
endrsubmit;
. . .
rsubmit taskn wait=no;
     /* code for parallel task #n   */
endrsubmit;
waitfor _all_ task1 task2 . . . taskn;
signoff _all_;
```

For more information, see "GRDSVC_ENABLE Function" on page 261.

For detailed syntax information, see *SAS/CONNECT User's Guide*.

Using Python with a SAS Grid

The open-source SASPy Python module converts Python code to SAS code and runs the code in SAS. The module works in any SAS 9.4 grid environment. One option available in SASPy is to run the code on a workspace server that can be set up as a grid-launched workspace server if a SAS Grid is available. You can use the module in both line mode and batch Python, and well as in Jupyter Notebooks. The results include ODS output, and can be returned as Panda data frames.

To submit code to a SAS Grid environment using the SASPy module, specify the IOM connection method in your Python code. SAS Grid Manager or SAS Grid Manager for Platform automatically controls connections to the grid nodes and provides the standard grid monitoring and administrative functions. Not all Python methods are supported by SASPy, but because the module is open source, you can add or modify Python methods to the module.

To download SASPy, go to the GitHub repository.

Using SAS Studio with a SAS Grid

SAS Studio can take advantage of the processing capabilities of a SAS Grid. The approach to take to add grid support depends on the edition of SAS Studio that you are using.

With all editions of SAS Studio, you can use the SAS/CONNECT GRDSVC_ENABLE, SIGNON, and RSUBMIT functions in your code to send the code to the grid. Use the GRDSVC_ENABLE function to identify the SAS/CONNECT session that should be sent to the grid, specify that session name in the SIGNON statement, and then submit the code to the session in the RSUBMIT statement.

For example, you could use the following code to specify processing on the grid:

```
%let rc=%sysfunc(grdsvc_enable(gridsess,server=SASApp));
options autosignon;
rsubmit gridsess wait=no;
    /* code to be processed on the grid */
endrsubmit;
waitfor gridsess;
signoff gridsess;
```

See "SAS Functions for SAS Grid" on page 261 and the *SAS/CONNECT 9.4 User's Guide* for syntax information.

If you are using SAS Studio Enterprise Edition together with SAS Grid Manager or SAS Grid Manager for Platform, you can convert the SAS Studio workspace server to use grid load balancing. During the conversion process, you can choose to launch the workspace server either by the grid or by the object spawner on the least-used machine on the grid. After you convert the workspace server, the grid load-balanced server processes the SAS statements in SAS Studio. See "Using SAS Grid Manager or SAS Grid Manager for Platform for Server Load Balancing" on page 242 for more information.

In order to run code from any edition of SAS Studio on the grid, you must have SAS Grid Manager or SAS Grid Manager for Platform licensed. If you are using SAS Grid Manager for Platform, you must also have LSF (a component of Platform Suite for SAS) installed on the same machine as the SAS Studio workspace server.

If you are using version 3.1 of SAS Studio, modify the SAS Studio property metadata to specify that the application is grid capable. Identifying SAS Studio as grid capable enables you to associate grid options sets with the application. Grid options sets are collections of grid options, SAS options, and required resources that you can map to users and applications (in this case, SAS Studio). For more information, see Chapter 14, "Working With Grid Options Sets," on page 247.

To specify that SAS Studio is grid capable, use SAS Management Console to add the keyword **`isGridCapable:*SAS Studio Mid-Tier*`** to the metadata for the SAS Studio folder. See "Specifying That an Application Is Grid Capable" on page 251 for complete instructions. If you are using version 3.2 or later of SAS Studio, you do not need to modify the metadata.

Using SAS Enterprise Guide and SAS Add-In for Microsoft Office with a SAS Grid

Types of Grid Enablement

Jobs generated by SAS Enterprise Guide and SAS Add-In for Microsoft Office can take advantage of a SAS grid using one or a combination of these approaches:

- Using server-side load balancing through a grid-launched workspace server. When SAS Enterprise Guide or SAS Add-In for Microsoft Office request the use of a workspace server to run a job, the object spawner launches a new workspace server on the grid on a node that meets the criteria of the queue to which the job was submitted. The job then runs on the workspace server.See "Using SAS Grid Manager or SAS Grid Manager for Platform for Server Load Balancing" on page 242 for more information about specifying grid load balancing on a workspace server.
- Using server-side load balancing through a grid load-balanced workspace server. When SAS Enterprise Guide or SAS Add-In for Microsoft Office request the use of a workspace server to run a job, the object spawner uses SAS Grid Manager or SAS Grid Manager for Platform to automatically launch the server on the least busy node on the grid. The code from SAS Enterprise Guide or SAS Add-In for Microsoft Office is then run on the workspace server. See "Using SAS Grid Manager or SAS Grid Manager for Platform for Server Load Balancing" on page 242 for more information about specifying grid load balancing on a workspace server.
- Using SAS Enterprise Guide or SAS Add-In for Microsoft Office's ability to enable projects or tasks to run on the grid. If you are using SAS Enterprise Guide or SAS Add-In for Microsoft Office 5.1 or later, the code for the project or the task is wrapped in RSUBMIT/ENDRSUBMIT statements and automatically sent to the SAS grid for processing. To specify this type of processing, perform one of the following actions:
  - Specify **Use grid if available** in the Project Properties window or the Task Properties window.
  - Specify the **Initialize Grid (if available)** option when connecting to a workspace in the Options dialog box under SAS Programs.
  - Set the EGGridPolicy or the AMOGridPolicy extended attribute on the logical grid server to a value of *Force*. See "Using Extended Attributes to Specify Grid Usage" on page 232 for more information about these attributes.

  Note: These options are not needed if you use grid-launched workspace servers.

  If you are using grid-launched workspace servers and do not want the **Use grid if available** option to be available, set the EGGridPolicy or the AMOGridPolicy to a value of *Ignore*. The *Ignore* value is available in the first maintenance release after version 6.1 of SAS Enterprise Guide and SAS Add-In for Microsoft Office.
- Using SAS Enterprise Guide or SAS Add-In for Microsoft Office's ability to insert custom code before and after tasks and SAS programs to route the processing to the grid. If you are using SAS Enterprise Guide 4.3 or earlier, you can wrap the code for the project or the task in RSUBMIT/ENDRSUBMIT statements through the use of custom code options that are inserted before and after the code that is submitted with the program. You must also modify the application servers, the SAS Enterprise Guide configuration files, and the SAS Add-In for Microsoft Office options.

  Note: This option is not needed if you use grid-launched workspace servers.

If you use grid-launched workspace servers, you should not specify any of the other grid-related options. If you are using the first maintenance release of version 6.1 of SAS Enterprise Guide or SAS Add-In for Microsoft Office, you can prevent users

from using the grid options by specifying a value of *Ignore* for the EGGridPolicy or the AMOGridPolicy extended attribute on the logical grid server. See "Using Extended Attributes to Specify Grid Usage" on page 232 for more information about these attributes.

If you are not using grid-launched workspace servers, you must specify grid load balancing options, and you might have to wrap your code in RSUBMIT/ENDRSUBMIT statements. The statements are required if you are using SAS Enterprise Guide or SAS Add-In for Microsoft Office 4.3 or earlier. If you are using SAS Enterprise Guide or SAS Add-In for Microsoft Office 5.1 or later, the statements are not required.

Parallel Execution and Grid Enablement

The option **Allow parallel execution on the same server** is not an option to control processing on the grid, but it can affect how grid-enabled jobs are processed. The option causes SAS Enterprise Guide to create a workspace server for each parallel process in a project, regardless of whether the project is grid-enabled or not. The following list explains how this option interacts with the **Use grid if available** option:

Neither option is selected
: All code runs in a single workspace server.

Only **Use grid if available** option selected
: A workspace server starts, performs a grid-enabled SIGNON, and submits code to the grid.

Only **Allow parallel execution** option selected
: Multiple workspace servers are started, one for each parallel code path in a project.

Both options selected
: Multiple workspace servers are started. Each server signs on to the grid. Each parallel code path in a project is submitted to the grid.

Selecting both options might cause problems because the GRIDWORK and RMTWORK libraries are duplicated in each workspace server. This causes problems for SAS Enterprise Guide when it tries to read from one of those libraries. Selecting the **Allow parallel execution** option might cause problems because one execution stream might need to use or write to a data set that is in use by another stream.

If you are using grid-launched workspace servers, you do not need to enable the **Use Grid if available** option because the workspace server for SAS Enterprise Guide is already running on the grid. The **Allow parallel execution** option can improve run-time efficiency, but it is more complex because it can run multiple and separate SAS sessions. When possible, perform debug and testing operations with the **Allow parallel execution** option disabled.

Using Extended Attributes to Specify Grid Usage

You can use the EGGridPolicy and AMOGridPolicy attributes on the logical grid server definition to control whether SAS Enterprise Guide or SAS Add-In for Microsoft Office version 5.1 or later uses the grid. These attributes override any

other settings in the applications. Here are the possible values for the attributes (values are not case-sensitive):

Noforce
: jobs are sent to the grid depending on the setting of the option **Use grid if available**.

Force
: jobs are always sent to the grid, regardless of the value of the option **Use grid if available**.

Ignore
: jobs are never sent to the grid, regardless of the value of the option **Use grid if available**. This value is supported for the first maintenance release after version 6.1 of SAS Enterprise Guide and SAS Add-In for Microsoft Office.

Assigning Libraries in a Grid

In SAS 9.2 and later versions, SAS sessions on the grid use the METAAUTORESOURCES option by default. This option causes SAS libraries that are defined in metadata and identified as “pre-assigned” to automatically be assigned when the SAS session is started. Using pre-assigned libraries with the METAAUTORESOURCES option ensures that the libraries used in the code generated by SAS Enterprise Guide and SAS Add-In for Microsoft Office are available to the SAS sessions on the grid.

However, if your programs use a large number of libraries, you might not want to make all of these libraries pre-assigned. Automatically assigning a large number of libraries could cause performance problems, and not all libraries are likely to be used for all programs. To minimize the performance overhead, define the libraries in SAS metadata but do not identify them as pre-assigned. When you need to refer to the library, you can then use a LIBNAME statement using the META LIBNAME engine.

Developing SAS Programs Interactively Using a Grid

Maintaining a Connection to the Grid

By default, when you start SAS Enterprise Guide or SAS Add-In for Microsoft Office, it connects to a single workspace server and keeps that connection active for the length of the session. If you interactively develop programs in SAS Enterprise Guide or SAS Add-In for Microsoft Office by highlighting and submitting lines of code, the codes use items such as libraries, WORK files, and SAS global statements on the workspace server.

If you are submitting grid-enabled code from SAS Enterprise Guide or SAS Add-In for Microsoft Office in a grid environment, then items such as libraries and SAS global statements must be accessed through the grid rather than through a single

workspace server. To maintain access to these items, you must maintain a connection to the grid while you are developing programs interactively. If you are using grid-launched workspace servers, this is not an issue.

Managing Interactive Workload

When SAS Enterprise Guide or SAS Add-In for Microsoft Office is used for interactive program development, the workload is likely to consist of short bursts of work interspersed with varying periods of inactivity while the user considers their next action. The SAS grid configuration can best support this scenario with these configuration settings:

- Increase the number of job slots for each machine.

  Increasing the number of job slots increases the number of simultaneous SAS sessions on each grid node. Because the jobs that are run on the grid are not I/O or compute intensive like large batch jobs, more jobs can be run on each machine.

- Implement CPU utilization thresholds for each machine.

  If all users submit CPU-intensive work at the same time, SAS Grid Manager or SAS Grid Manager for Platform can suspend some jobs and resume the suspended jobs when resources are available. This capability prevents resources from being overloaded.

The following example shows a sample LSB.HOSTS file that is configured with job slots set to 32, a scheduling threshold set to 70% CPU utilization, and a suspension threshold set to 80% CPU utilization. The settings needed for a specific site depend on the number of users and the size of the grid nodes.

```
HOST_NAME MXJ    ut        r1m     pg    ls    tmp  DISPATCH_WINDOW  #Keywords
#default   !     ()         ()     ()    ()     ()      ()            #Example
host01     32    0.7/0.8    ()     ()    ()     ()      ()            # host01
host02     32    0.7/0.8    ()     ()    ()     ()      ()            # host02
host03     32    0.7/0.8    ()     ()    ()     ()      ()            # host03
host04     32    0.7/0.8    ()     ()    ()     ()      ()            # host04
host05     32    0.7/0.8    ()     ()    ()     ()      ()            # host05
End Host
```

Using SAS Stored Processes with a SAS Grid

SAS stored processes can take advantage of a SAS grid by using server-side load balancing through a stored process server. You can convert a stored process server to use load balancing, and then specify grid as the balancing algorithm. When you send jobs from a SAS stored process to this server, the server automatically sends the job to the least busy node on the grid. See “Using SAS Grid Manager or SAS

Grid Manager for Platform for Server Load Balancing" on page 242 for more information about specifying grid load balancing on a stored process server.

Using SAS Data Integration Studio with a SAS Grid

Scheduling SAS Data Integration Studio Jobs on a Grid

You can schedule jobs from within SAS Data Integration Studio and have those jobs run on the grid. You deploy the job for scheduling in SAS Data Integration Studio, and then use the Schedule Manager plug-in in SAS Management Console to specify the schedule and the scheduling server. For more information, see "Scheduling Jobs on a Grid" on page 227. Also see *Scheduling in SAS*.

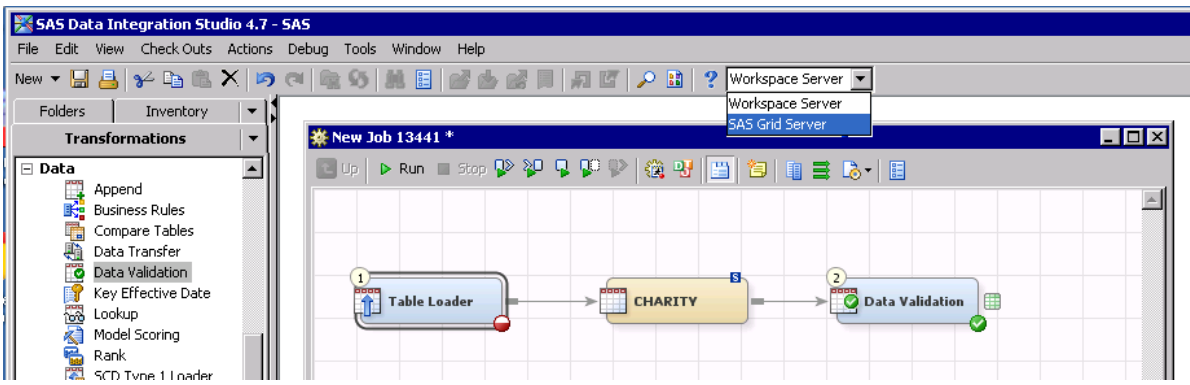
Multi-User Workload Balancing with SAS Data Integration Studio

SAS Data Integration Studio enables users to directly submit jobs to a grid or to a workspace server that is launched by the grid. This capability allows the submitted jobs to take advantage of load balancing and job prioritization that you have specified in your grid. SAS Data Integration Studio also enables you to specify the workload that submitted jobs should use. This capability enables users to submit jobs to the correct grid partition for their work.

To submit a job to the grid, select the SAS Grid Server component in the **Server** menu on the Job Editor toolbar. Click **Submit** in the toolbar to submit the job to the grid.

This step is not needed if you use grid-launched workspace servers.

Figure 13.1 Submitting a Job to the Grid



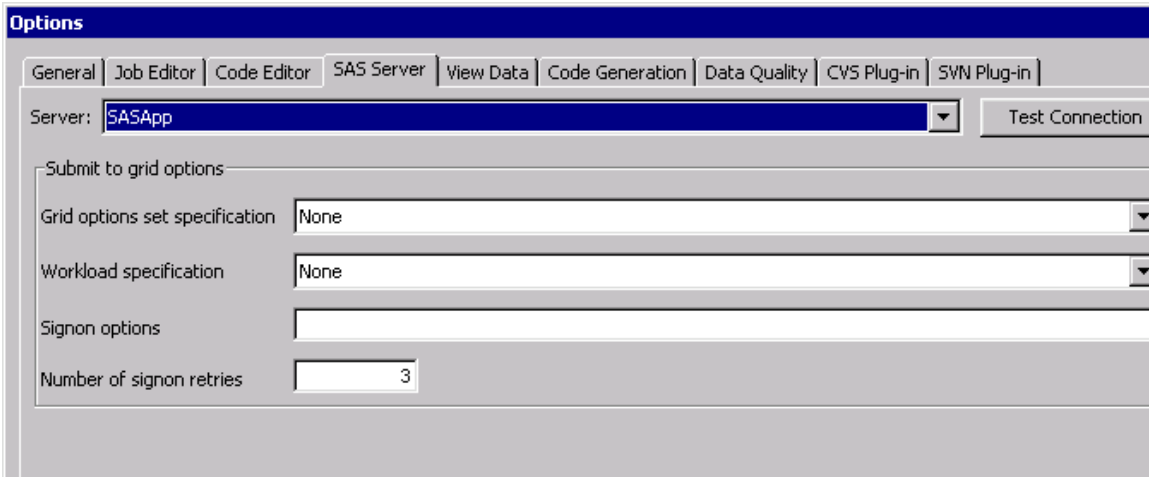
To specify a workload value for the server, follow these steps:

1 On the **SAS Data Integration Studio** menu bar, select **Tools** ⇨ **Options**, and then select the **SAS Server** tab on the Options dialog box.

2 Select the SAS grid server in the **Server** field.

3 Select the workload to use for the submitted jobs in the **Workload specification** field.

Figure 13.2 Selecting the Workload



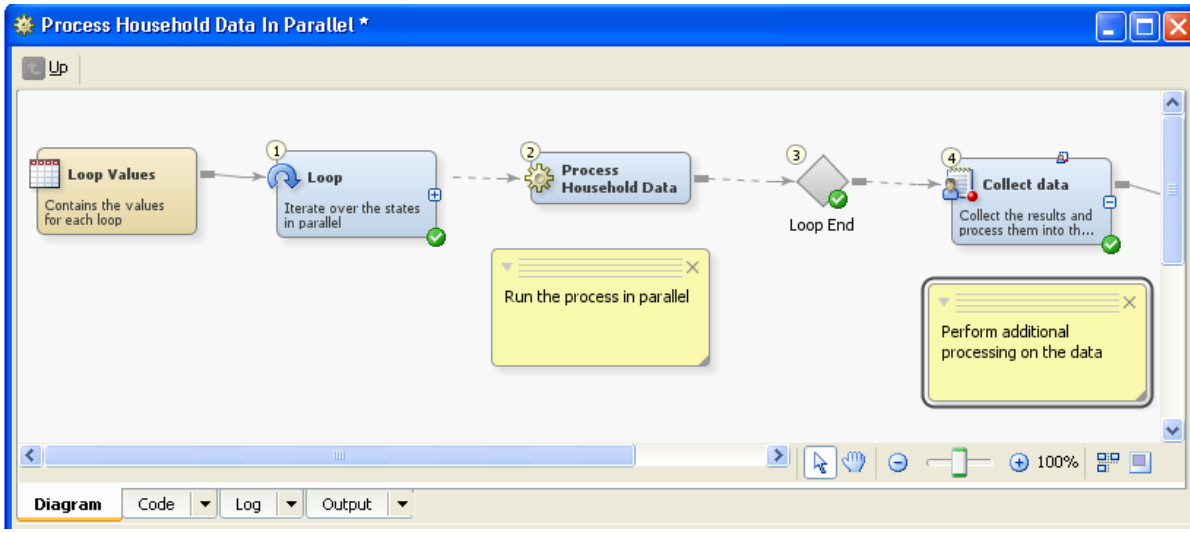
4 You can also specify the workload in a grid options set and specify the options set name in the **Grid options set specification** field. Only grid options sets that are mapped to SAS Data Integration Studio are listed in this field. For more information, see "Modifying SAS Logical Grid Server Definitions" on page 210.

SAS Grid Manager uses the workload value to send the submitted job to the appropriate grid partition. For more information about the other steps required, see "Defining and Specifying Resources" on page 124.

Parallel Workload Balancing with SAS Data Integration Studio

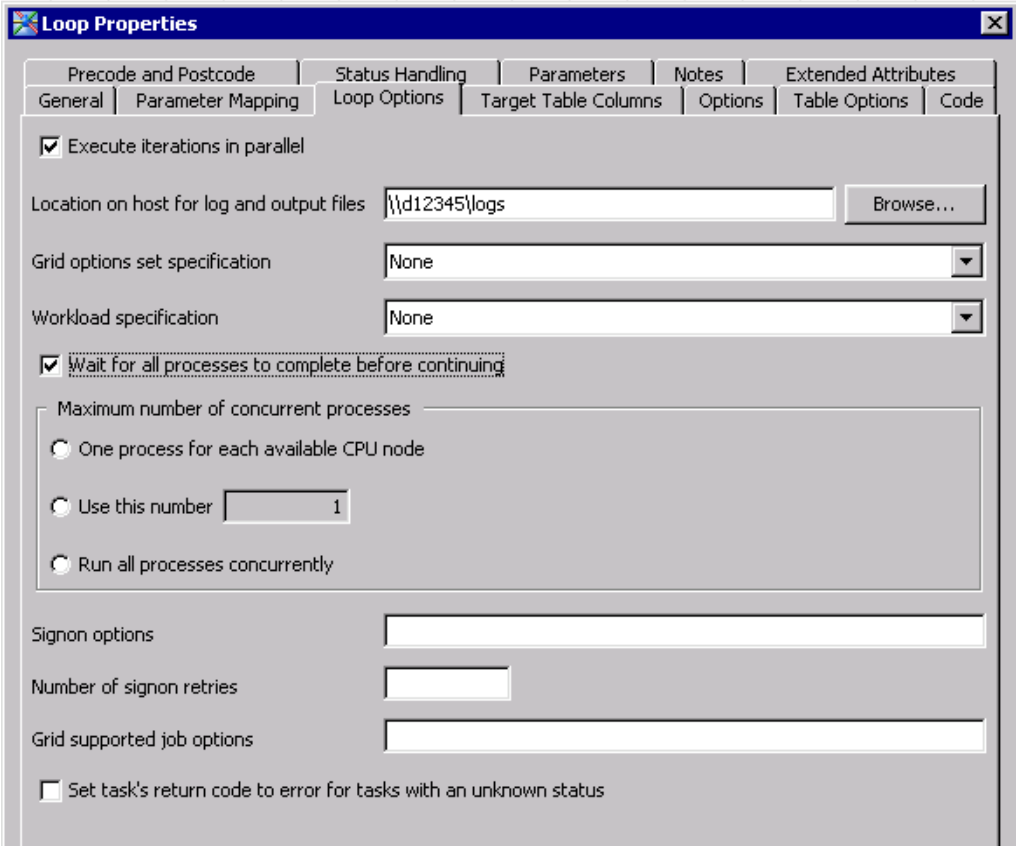
A common workflow in applications created by SAS Data Integration Studio is to repeatedly execute the same analysis against different subsets of data. Rather than running the process against each table in sequence, use a SAS grid environment to run the same process in parallel against each source table, with the processes distributed among the grid nodes. For this workflow, the Loop and Loop-End transformation nodes can be used in SAS Data Integration Studio to automatically generate a SAS application that spawns each iteration of the loop to a SAS grid via SAS Grid Manager or SAS Grid Manager for Platform.

Figure 13.3 *Loop and Loop-End Transformation Nodes*



To specify options for loop processing, open the Loop Properties window and select the **Loop Options** tab. You can specify the workload for the job, and you can specify how many processes can be active at once.

Figure 13.4 *Loop Properties Dialog Box*



For more information, see *SAS Data Integration Studio: User's Guide*.

Specifying Workload for the Loop Transformation

A SAS Data Integration Studio user performs these steps to specify an LSF resource in the properties for a Loop Transformation in a SAS Data Integration Studio job. When the job is submitted for execution, it is submitted to one or more grid nodes that are associated with the resource.

It is assumed that the default SAS Application Server for SAS Data Integration Studio has a Logical SAS Grid Server component, which was updated in the metadata repository. For more information, see "Defining and Specifying Resources" on page 124.

1. In SAS Data Integration Studio, open the job that contains the Loop Transformation to be updated.
2. In the Process Designer window, right-click the metadata object for the Loop Transformation and select **Properties**.
3. In the Properties window, click the **Loop Options** tab.
4. On the **Loop Options** tab, in the **Grid options set specification** text box, select the name of the desired grid options set. The entry is case sensitive.
5. Click **OK** to save your changes, and close the Properties window.

Using SAS Enterprise Miner with a SAS Grid

There are three cases where SAS Enterprise Miner uses a SAS grid:

- during model training, for parallel execution of nodes within a model training flow
- during model training, for load balancing of multiple flows from multiple data modelers
- during model scoring, for parallel batch scoring

The workflow for SAS Enterprise Miner during the model training phase consists of executing a series of different models against a common set of data. Model training is CPU-intensive and I/O-intensive. The process flow diagram design of SAS Enterprise Miner lends itself to processing on a SAS grid because each model is independent of the other models. SAS Enterprise Miner generates the SAS program to execute the user-created flow and also automatically inserts the syntax needed to run each model on the grid. Because the models can execute in parallel on the grid, the entire process is accelerated.

In addition, SAS Enterprise Miner is typically used by multiple users who are simultaneously performing model training. Using a SAS grid can provide multi-user load balancing of the flows that are submitted by these users, regardless of whether the flows contain parallel subtasks.

The output from training a model is usually Base SAS code that is known as scoring code. The scoring code is a model, and there are usually many models that need to be scored. You can use SAS Grid Manager or SAS Grid Manager for Platform to score these models in parallel. This action accelerates the scoring process. You can use any of these methods to perform parallel scoring:

- Use the SAS Grid Manager Client Utility to submit each model independently to the SAS grid.
- Use the Schedule Manager plug-in to create a flow that contains multiple models and schedule the flow to the SAS grid. Because each model is independent, the models are distributed across the grid when the flow runs.
- Use SAS Data Integration Studio to create a flow to loop multiple models, which spawns each model to the grid.

Starting with SAS Enterprise Miner 6.2, you can specify that projects are sent to the grid by default.

In SAS Enterprise Miner 12.3, specify default grid processing by changing the attributes on the logical workspace server. Follow these steps:

1. In SAS Management Console, expand the Server Manager plug-in and then expand the SAS Application Server (typically named SASApp).
2. Select the Logical Workspace Server and select **File** ⇨ **Properties**.
3. On the **Extended Attributes** tab, click **New**.

4 In the **Field Name** column, enter `EM_USE_GRID_IF_AVAILABLE`.

5 In the **Value** column, enter `Y`.

6 Click **OK** to save the property changes.

Setting the `EM_USE_GRID_IF_AVAILABLE` attribute to `Y` causes all of the SAS Enterprise Miner projects to execute their diagrams with grid processing enabled. Changing the option does not affect any previously launched Enterprise Miner clients, but affects all new clients that are started after the change is made.

You can disable grid processing for all SAS Enterprise Miner clients by setting the value of the `EM_USE_GRID_IF_AVAILABLE` attribute to `N`.

SAS Enterprise Miner projects also have a grid processing project preference that persists with the project metadata. This setting is available if the `EM_USE_GRID_IF_AVAILABLE` attribute is not defined or if it is set to `U`. This project preference is ignored and disabled if the `EM_USE_GRID_IF_AVAILABLE` attribute is set to `Y` or `N`. Follow these steps to modify the preference in the SAS Enterprise Miner client application:

1 In the SAS Enterprise Miner client, open the project.

2 Select **Options** ⇨ **Preferences** to display the Preferences dialog box.

3 Select the **Use grid processing when available** option from the menu that is associated with the **Grid Processing** field. This field is in the **Run Options** area of the dialog box.

4 Click **OK** to save the property changes.

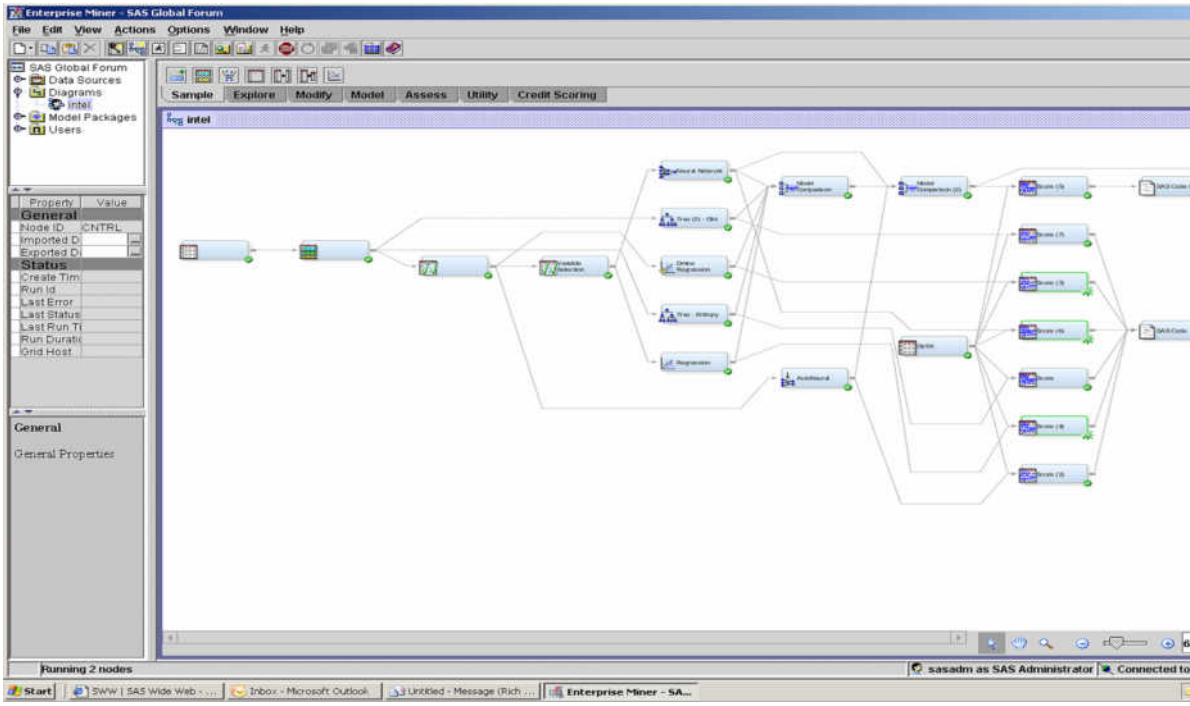
To disable grid processing, select **Never use grid processing** from the **Grid processing** field.

Beginning with SAS Enterprise Miner 14.2, you can see the value of the workspace server extended attribute `EM_USE_GRID_IF_AVAILABLE` in the SAS Enterprise Miner Preferences dialog box. The value is displayed in the **Run Options** section in the **Workspace Server Grid Attribute** property. However, it can be modified only by an administrator in SAS Management Console.

Note: When you change the `EM_USE_GRID_IF_AVAILABLE` attribute, the change is not immediately applied to SAS Enterprise Miner client sessions that have already read and cached the settings for the affected logical workspace server.

In SAS Enterprise Miner 6.2 to 12.1, specify default grid processing by changing a configuration property. To set this property, locate the file app.config. A typical location for this file is `C:\SAS\Config\AnalyticsPlatform\apps\EnterpriseMiner\app.config`. Locate the property em.enablegrid. To turn on default grid processing, specify a value of `Y`. If you specify a value of `N`, you must send projects individually to the grid.

Figure 13.5 Grid Processing with SAS Enterprise Miner



Using SAS Risk Dimensions with a SAS Grid

The iterative workflow in SAS Risk Dimensions is similar to that in SAS Data Integration Studio. Both execute the same analysis over different subsets of data. In SAS Risk Dimensions, the data is subsetted based on market states or by instruments. Each iteration of the analysis can be submitted to the grid using SAS Grid Manager or SAS Grid Manager for Platform to provide load balancing and efficient resource allocation.

Because every implementation is different, an implementation of SAS Risk Dimensions in a grid environment must be customized to your specific business and data requirements.

Using SAS Grid Manager or SAS Grid Manager for Platform for Server Load Balancing

Overview

SAS servers, such as SAS workspace servers and SAS OLAP 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 or SAS Grid Manager for Platform installed, you can configure workspace servers and other SAS servers to use SAS Grid Manager or SAS Grid Manager for Platform in order to provide load balancing. Because SAS Grid Manager or SAS Grid Manager for Platform accounts for all of the resource consumption on the machine, SAS Grid Manager or SAS Grid Manager for Platform can make better decisions about which machine is the best candidate for a server session.

SAS Grid Manager or SAS Grid Manager for Platform can provide load-balancing capabilities for these types of servers:

- workspace servers
- pooled workspace servers
- stored process servers
- OLAP servers

Any SAS product or solution that uses workspace servers, pooled workspace servers, stored process servers, or OLAP servers would benefit from using SAS Grid Manager or SAS Grid Manager for Platform to provide load balancing. This includes (but is not limited to) SAS Enterprise Guide, SAS Studio Enterprise Edition, SAS Data Integration Studio, SAS Enterprise Miner, SAS Web Report Studio, and SAS Marketing Automation. However, using the grid to provide load balancing also increases overhead, so your jobs might take longer to run.

You can use two methods to provide workspace server load balancing through SAS Grid Manager or SAS Grid Manager for Platform. You can have SAS Grid Manager or SAS Grid Manager for Platform start the workspace servers, or you can have the object spawner start the servers.

Load Balancing Methods

Grid-Launched Servers

With grid-launched workspace servers, the spawner passes the request on to SAS Grid Manager or SAS Grid Manager for Platform when an application requests a SAS workspace server from the object spawner. SAS Grid Manager or SAS Grid Manager for Platform submits the job to the grid provider, which determines the best server to run the job based on the policy of the queue. The grid provider uses factors such as CPU utilization, network and disk input and output, disk space, available memory, and queue limits in determining which machine to use.

Because the grid-launched workspace servers are started by the grid provider, jobs on these servers appear as grid jobs.

If you are using SAS Grid Manager for Platform, an error will result if you use a grid-launched workspace server that specifies more hosts than the queue definition that you are using. For example, using a queue definition that specifies only host2 together with a logical workspace server definition that specifies host1, host2, and host3 causes an error. To accommodate this type of environment, add the line `ENABLE_HOST_INTERSECTION=Y` to the lsb.params file.

Servers Started by the Object Spawner

When an application or process (for example, SAS Enterprise Guide) requests a SAS server from the object spawner, the spawner determines which hosts can run the new servers. The spawner sends the list of hosts to SAS Grid Manager or SAS Grid Manager for Platform, which then determines which host is open and is the least busy. The object spawner then directs the client connection to a server on the least busy node on the grid. If there are multiple servers already running on the machine, then the spawner directs the client to the server with the fewest number of connections.

If you are using SAS Grid Manager for Platform and using the grid load balancing algorithm, jobs are not started by LSF. As a result, jobs that run on the load-balanced servers do not appear as LSF jobs.

For more information, see “Understanding Server Load Balancing” in *SAS Intelligence Platform: Application Server Administration Guide*.

SAS OLAP Servers

Because the SAS object spawner is not used to start SAS OLAP servers, the request is not processed through the object spawner if the client requests a SAS OLAP server. However, the same load-sharing logic is used to determine which

OLAP server is used for the client connection. As a result, the SAS client connections are spread around to all of the servers on the grid.

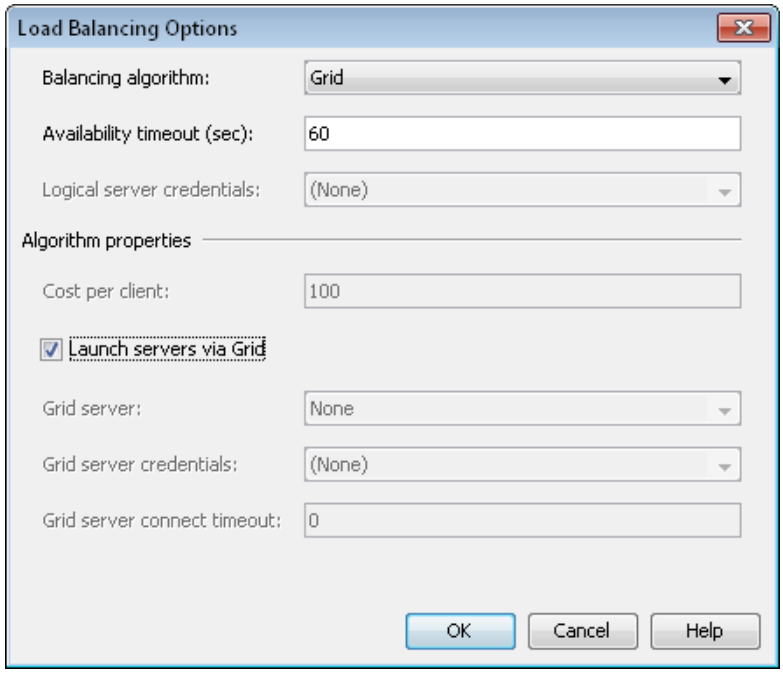
Converting Servers to Use Grid Load Balancing

To use SAS Grid Manager for load balancing for a workspace server or an OLAP server, follow these steps:

1 In the Server Manager plug-in in SAS Management Console, select the server that will use load balancing.

2 Select **Convert To** ⇨ **Load Balancing** from the **Actions** menu or the context menu for the logical server. The Load Balancing Options dialog box appears.

Figure 13.6 Server Load Balancing Options Dialog Box



3 Specify the following values:

Balancing algorithm
: Select **Grid**.

Launch Servers via Grid
: Select this check box to specify that SAS Grid Manager or SAS Grid Manager for Platform will launch the new servers. Deselect this check box if you want the object spawner to start the servers. This option is not available for OLAP servers.

Grid server
: Select a grid server that was defined during installation and configuration. This field is active only if you did not select **Launch Servers via Grid**.

Grid server credentials
: Select the credentials that the object spawner uses to authenticate to the grid server. This field is active only if you did not select **Launch Servers via Grid**.

4 Click **OK** to save your changes to the server metadata.

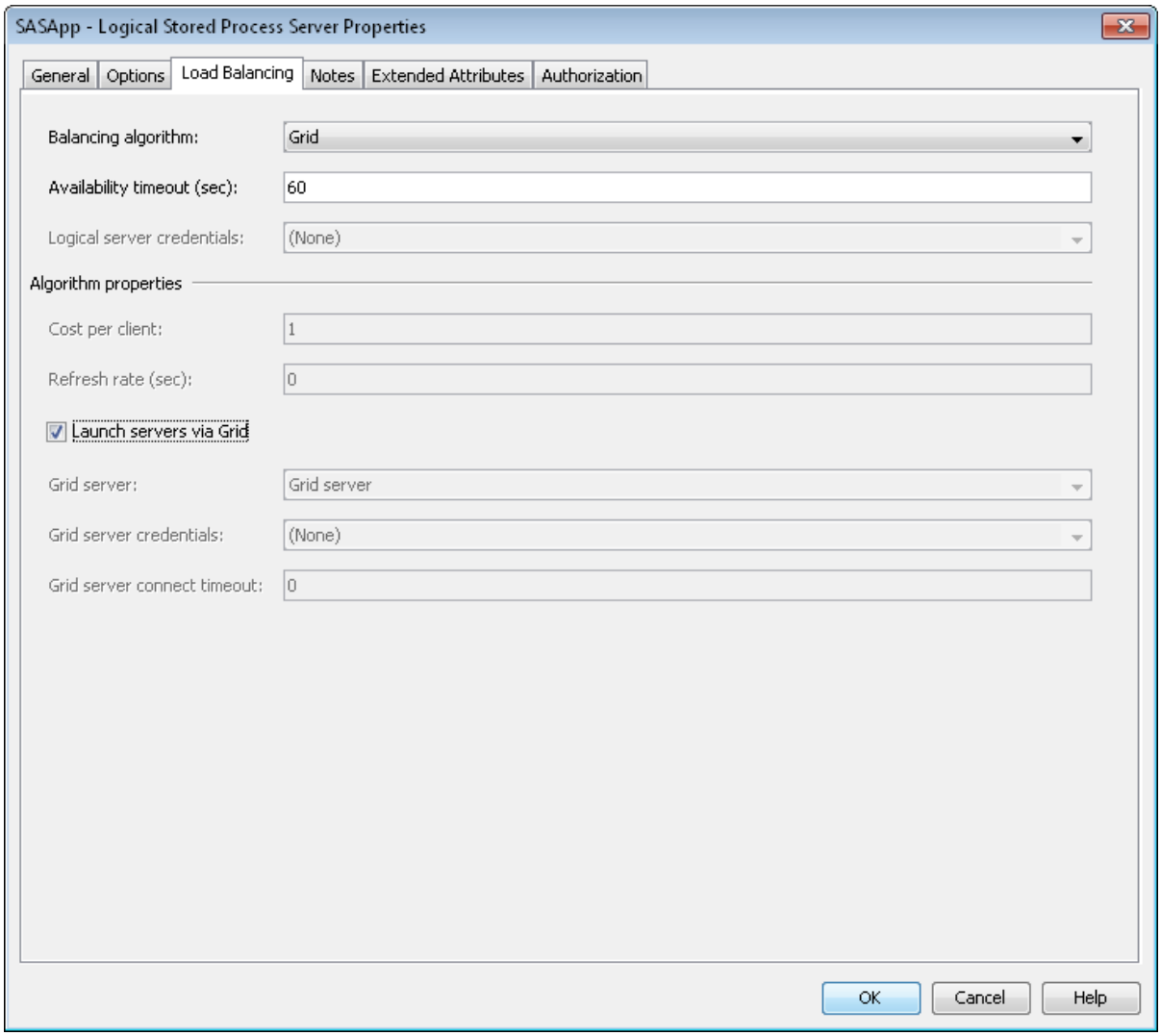
To use SAS Grid Manager or SAS Grid Manager for Platform for load balancing for a stored process server or a pooled workspace server, follow these steps:

1 In the Server Manager plug-in in SAS Management Console, select a logical server that will use load balancing.

2 Select **Properties** from the **Actions** menu or the context menu. The Properties dialog box appears.

3 Select the **Load Balancing** tab.

Figure 13.7 Load Balancing Options for a Server



4 Specify the following values:

Balancing algorithm
Select **Grid**.

Launch Servers via Grid
Select this check box to specify that SAS Grid Manager or SAS Grid Manager for Platform will launch the new servers. Deselect this check box if you want the object spawner to start the servers.

Grid server
: Select a grid server that was defined during installation and configuration. This field is active only if you did not select **Launch Servers via Grid**.

Grid server credentials
: Select the credentials that the object spawner uses to authenticate to the grid server. This field is active only if you did not select **Launch Servers via Grid**.

In addition to modifying these server definitions, you must also change server configuration files. For information about the changes that you need to make, see "Understanding Server Load Balancing" in *SAS Intelligence Platform: Application Server Administration Guide*.

14

Working With Grid Options Sets

Understanding Grid Options Sets

A grid options set is a collection of grid options, SAS options, and required resources. Grid options sets are then mapped to a particular SAS client application and one or more specific metadata identities. The purpose of a grid options set is to enable a SAS grid administrator to define a collection of options in SAS metadata that map to one or more SAS client applications. Those options and are automatically applied to the workload that is submitted to the grid based on the identity of the user accessing the client application. SAS client applications such as the SAS Grid Manager Client Utility have been enhanced to support grid options sets.

For example, consider a scenario where you have many SAS Data Integration Studio users. The users are classified as either “power users” or “ad hoc users.” The power users should always submit their jobs to the POWERUSER queue and use the SAS options MEMSIZE=2048M and SORTSIZE=512M. The ad hoc users should always submit their jobs to the DIS queue and use the default SAS invocation options.

To support this scenario, define the workspace servers to be launched using the grid. You could define a grid options set called DISPOWERUSER that contains the grid option QUEUE=POWERUSER and SAS options MEMSIZE=2048M and SORTSIZE=512M. You would then map this grid options set to the SAS Data Integration Studio application and to the metadata identities that you identify as power users. You could then define a second grid options set called DISADHOCUSER that contains only the grid option QUEUE=DIS. You would also map this grid options set to the SAS Data Integration Studio application, but to the SASUSERS group. When a SAS Data Integration Studio user submits a job to the

grid, the grid options set mapping identifies which grid options set to use, and applies the proper options for the job. This process is completely transparent to the user.

Merging Option Values

When you use a grid options set, its values are merged with the default option values for the logical grid server. The way the values are merged and evaluated differs according to the type of option:

SAS Options
: values in the grid options set are appended to the values from the **SAS Options** field on the logical grid server's Properties window. Because the options are evaluated from left to right, the value from the grid options set is evaluated last and therefore overrides any similar default SAS option.

  For example, you have a logical grid server with a SAS option set to "-memsize 256" and you create a grid options set with a SAS option value of "-memsize 0". When the job is executed on the grid, the SAS command line has the options "-memsize 256 -memsize 0" at the end of the command line. The option "-memsize 0" is evaluated last, so that is the value used when processing the job.

Required resources
: values in the grid options set are added to the values from the **Required Resources** field on the logical grid server's Properties window. When the job runs on the grid, all specified resources must be available.

  For example, you might have a logical grid server with "SASApp" specified in the **Required Resources** field and a grid options set with a required resource specified as "MySQLDB". When the grid provider selects a host on which to run the job, it must select a host that has both "SASApp" and "MySQLDB" as resources associated with the host.

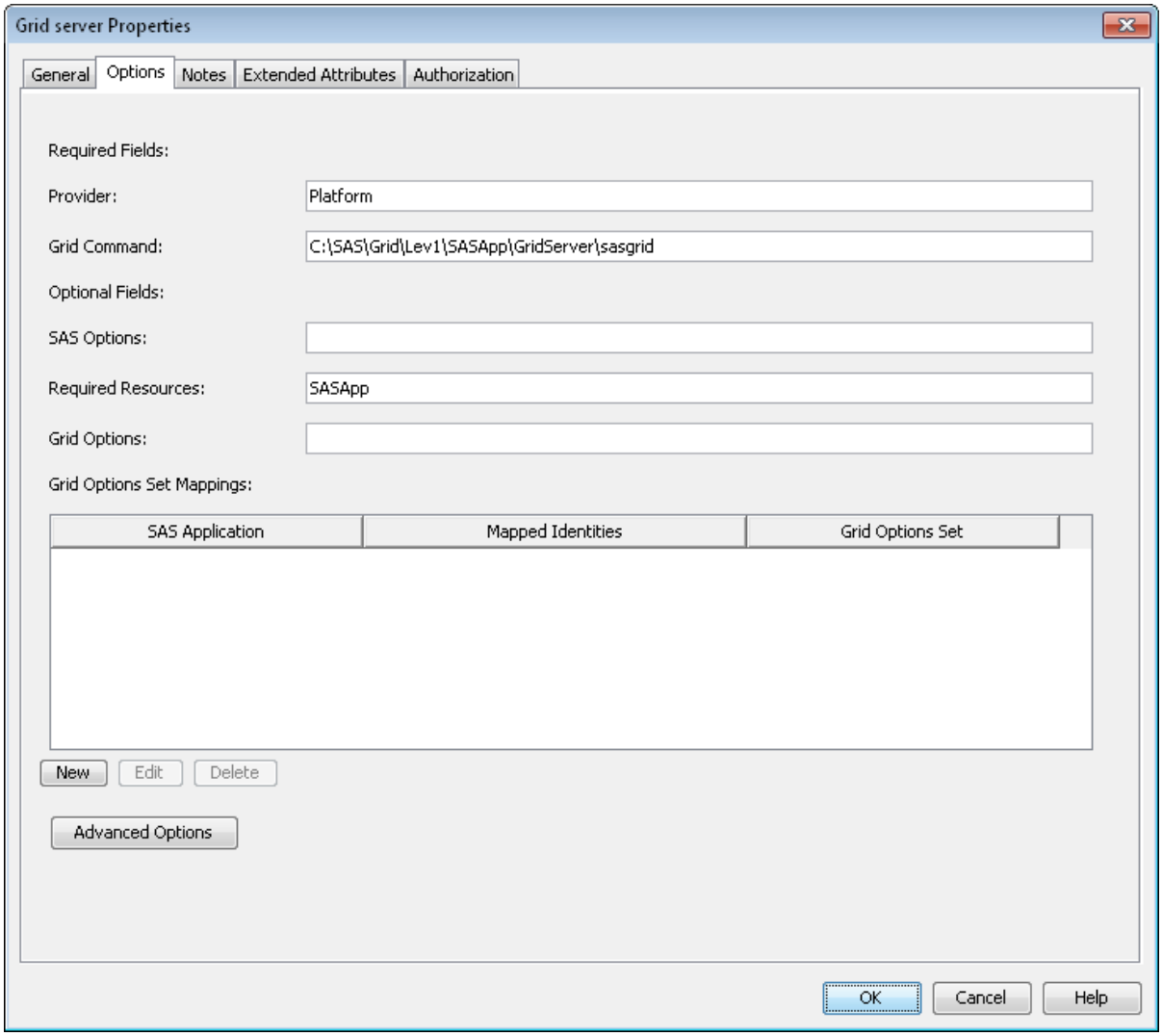
Grid options
: values in the grid options set are appended to the values from the **Grid Options** field on the logical grid server's Properties window. Because the options are evaluated from left to right, the value from the grid options set is evaluated last and therefore overrides any similar default grid option.

  For example, if you are using Platform Suite for SAS, you might have a logical grid server with a **Grid Options** value of "queue=normal" and you create a grid options set with a value of "queue=priority". When the job is passed to the grid provider module, the option string that is passed include the options "queue=normal queue=priority". Because the option "queue=priority" appears after "queue=normal", the priority queue is used for the job.

Create a Grid Options Set

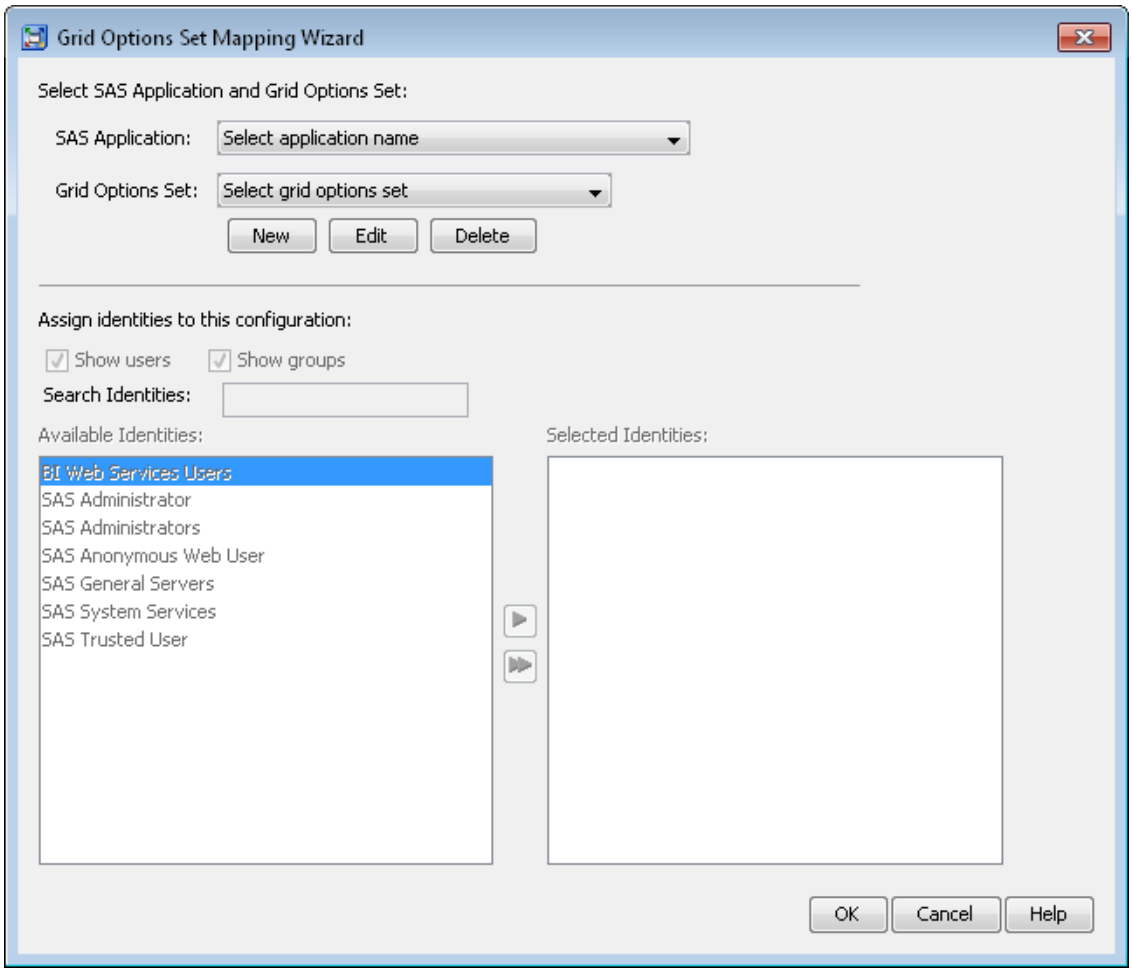
Grid options sets are created in SAS Management Console as part of a Grid Server definition. You must be a member of the SAS Administrators group in order to create a grid options set. To create a grid options set, follow these steps:

1. In SAS Management Console, expand **Server Manager**.
2. Expand a **Logical Grid Server** and select a **Grid Server** component.
3. Right-click the **Grid Server** component and select **Properties**.
4. In the Properties window, select the **Options** tab. The Grid Options Set Mappings table lists any existing mappings.



5. Click **New** to display the Grid Options Set Mapping Wizard dialog box.

The **New**, **Edit**, and **Delete** buttons are available only if you are a member of the SAS Administrators group.



6 From the **SAS Application** list box, select the SAS application to associate with this grid options set.

The applications that are listed in this field are those that have been identified as grid capable either during deployment or by specifying the isGridCapable keyword in the application's properties.

The options in the grid options set can be used when the workspace server starts if you specify the following criteria:

- You select an application that launches a workspace server.
- You specify that load balanced servers should be launched by the grid, rather than by the spawner.
- The application name is matched with the isGridCapable keyword. See "Specifying That an Application Is Grid Capable" on page 251 for more information.

7 Click **New** under the **Grid Options Set** list box to display the New Grid Options Set window.

8 In the **Grid Options Set Name** field, enter a name for the new grid options set.

9 Specify values in one or more of the following fields: **SAS Options**, **Required Resources**, or **Grid Options**. For information about these values and how they are applied, see "Merging Option Values" on page 248.

10 Click **OK** to save the grid options set.

11 In the **Identities** area, select identities (users or groups) to associate with the grid options set. You must select at least one identity to associate with the mapping. If no identity is selected, the mapping is not saved.

12 Click **OK** to save the grid options set mapping.

13 Click **OK** to save the options on the **Options** tab of the Properties window.

Specifying That an Application Is Grid Capable

The only applications that you can associate with a grid options set are ones that have been identified as being grid capable. Applications are identified as grid capable either by the application's deployment process or by specifying the isGridCapable keyword through the application's property values.

To verify which applications have been identified as grid capable during deployment, you can define a grid options set mapping before you have set the isGridCapable keyword on any applications through the application properties. The applications that are listed as available for mapping are the ones that were identified as grid capable during deployment.

For all other applications that you want to map to a grid options set, you must manually identify each of them as grid capable. In order to apply the isGridCapable keyword to the correct application when running a job, SAS Grid Manager or SAS Grid Manager for Platform must be able to match these names:

- the application name as identified on the application folder
- the name that is sent by the application when the object spawner starts the workspace server

If the names do not match, then you can use wildcards to help match the names. Names that do not match might occur in either of the following situations:

- The application sends a different name from one invocation to the next.
- You have multiple versions of the same application that you want to map to the same grid options set.

These examples illustrate several scenarios:

- If the folder for an application is specified as "SAS Application A" and the application sends the name "SAS Application A", then the names match exactly and you can set the keyword to isGridCapable.
- If the folder for an application is specified as "SAS Application A" and the application sends the name "SAS Application A 4.6" some times and "SAS

Application A 4.7” at other times, then you should specify the keyword in the form isGridCapable:SAS Application A*.

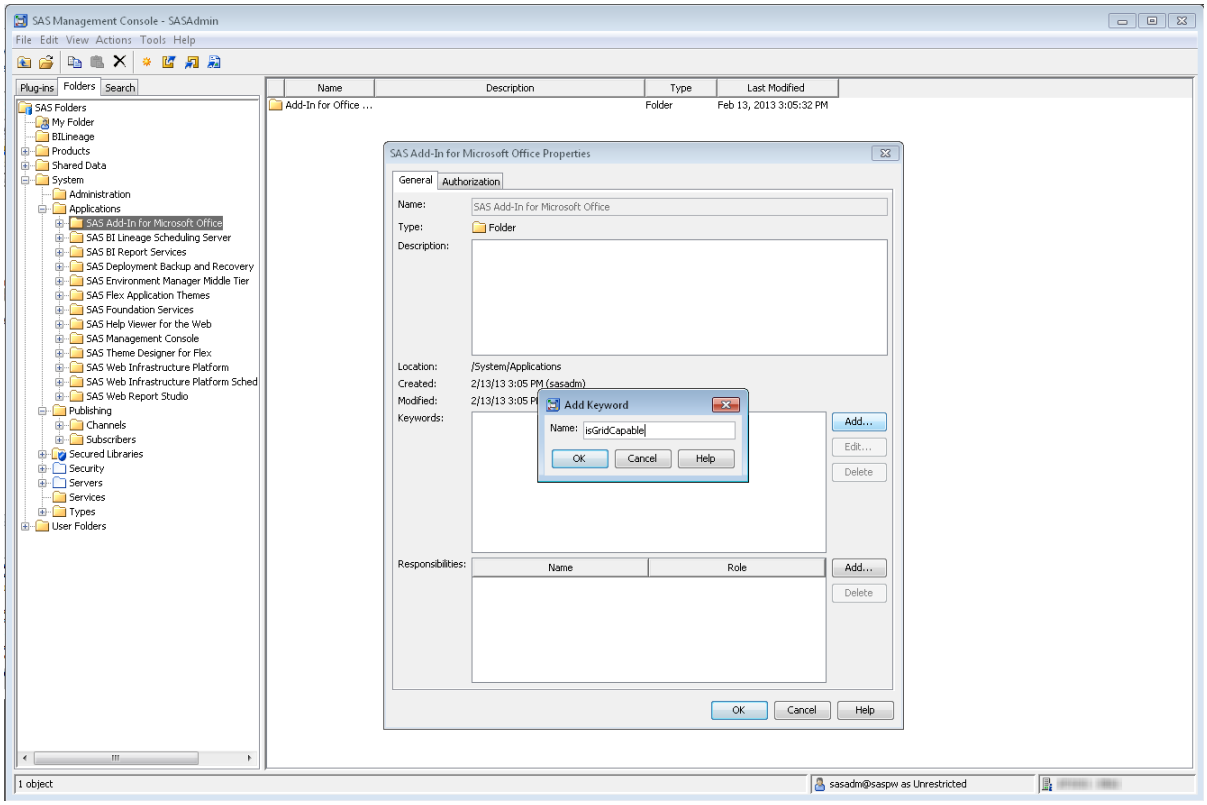
- If the folder for an application is specified as “SAS Application A” and the application sends the name “SAS Application A” some times and “Web Infrastructure - SAS Application A” at other times, then you should specify the keyword in the form isGridCapable:*SAS Application A.
- If the folder for an application is specified as “SAS Application A” and the application sends the name “Web Infrastructure - SAS Application A” some times and “SAS Application A 4.6” at other times, then you should specify the keyword in the form isGridCapable:*SAS Application A*.

For example, to make SAS Studio a grid-enabled application, you might specify the keyword as **`isGridCapable:*SAS Studio Mid-Tier*`**.

- The **`isGridCapable`** portion of the keyword enables SAS Studio Mid-Tier to be included in the list of grid-enabled applications.
- The **`*SAS Studio Mid-Tier*`** portion of the keyword enables the IOM application name as sent by SAS Studio to be matched to the SAS Studio Mid-Tier application folder so that grid options sets can be applied. The wildcards enable SAS Grid Manager or SAS Grid Manager for Platform to recognize the IOM name (Web Infra Platform Services 9.4 - SAS Studio Mid-Tier 3.1) and match it with the SAS Studio Mid-Tier options sets.

To set the isGridCapable keyword for an application, follow these steps:

1. In SAS Management Console, select the **Folders** tab.
2. Open the **System** ⇨ **Applications** folder.
3. Select the folder for the application whose property you want to change. Then, select **File** ⇨ **Properties**.
4. In the Properties window, click **Add** next to the **Keyword** area.
5. In the Add Keyword dialog box, type **`isGridCapable`** or **`isGridCapable:`***`application_name`* with wildcards as needed in the **Name** field.



6 Click **OK** to close the Add Keyword dialog box and then the Properties window.

After applications are identified as being grid capable and grid options sets are mapped to those applications, the options in the grid options set are used to start any grid-launched servers. See "Grid-Launched Servers" on page 243 for more information.

15

Restarting Jobs

Overview

An essential component of a highly available grid is the ability to handle SAS jobs that fail or have to be restarted for some reason. If a long-running job fails, it can cause a significant loss of productivity. After the failure is noticed, you must manually resubmit the job and wait while the program starts over again from the beginning. For SAS programs that run for a considerable amount of time, this can cause unacceptable delays.

The SAS Grid Manager Client Utility, combined with the SAS checkpoint restart feature, provides the capability to restart a job from the last successful job step.

If you are using SAS Grid Manager for Platform, the SAS Grid Manager Client Utility, combined with LSF queue policies, also provides the ability to set up a special queue to automatically send failed jobs to another host on the grid for continued execution.

Using SAS Checkpoint and Label Restart

The SAS Grid Manager Client Utility includes options that enable you to restart SAS programs from the last successful SAS procedure or DATA step. When the program runs, it records information about the SAS procedures and DATA steps or labels in the program and tracks the ones that have been passed during execution.

If the program fails and has to be restarted, SAS first executes global statements and macros. Then, it reads the checkpoint or label library to determine which checkpoints or labels have been passed. When SAS determines where the program stopped, execution is resumed from that point. Program steps that have already successfully completed are not executed again.

The restart capability is available on the grid only if you are using the SAS Grid Manager Client Utility for scheduling grid jobs. It is not available if you are using other application interfaces to submit work to the grid.

When you use the restart options, your SAS WORK library is relocated to shared storage in the job directory of the grid shared directory. Because this capability adds some overhead to your SAS program, it is not recommended for every SAS program that you run.

To set up the checkpoint or label restart capability, use the SAS Grid Manager Client Utility to submit the SAS program to the grid. Specify either the GRIDRESTARTOK argument (for checkpoints) or the GRIDLRESTARTOK argument (for labels). You cannot specify both arguments.

When you use the GRIDRESTARTOK argument, these options are automatically added to your SAS program:

STEPCHKPT
: enables checkpoint mode and causes SAS to record checkpoint-restart data.

STEPRESTART
: enables restart mode, ensuring that execution resumes at the proper checkpoint.

When you use the GRIDLRESTARTOK argument, these options are automatically added to your SAS program:

LABELCHKPT
: enables checkpoint mode for labeled code sections.

LABELRESTART
: enables restart mode, ensuring that execution resumes at the proper labeled section.

Other options are automatically added to control restart mode. See Checkpoint Mode and Restart Mode in *SAS Language Reference: Concepts* for a list of options and their definitions as well as complete information about enabling checkpoint restart mode in your SAS programs.

If the host that is running the job becomes unresponsive, the program is automatically restarted at the last checkpoint.

Setting Up Automatic Job Requeuing on SAS Grid Manager for Platform

If you are using SAS Grid Manager for Platform, you can set up a queue that automatically requeues and redispatches any job that ends with a specified return code or terminates due to host failure. Job requeuing enables you to handle situations where the host or the system fails while the job is running. Using the

requeue capability ensures that any failed jobs are automatically dispatched to another node on the grid.

To use this functionality on a SAS Grid Manager for Platform grid, you must use the SAS Grid Manager Client Utility and configure the SAS WORK library to run on shared storage.

To set up a queue for automatic restart, follow these steps:

1. Create a queue, including these two options in the queue definition:
   - `REQUEUE_EXIT_VALUES=`*`return_code_areturn_code_b ...return_code_n`* option in the queue definition. The *return_code* values are the job exit codes that you want to filter. Any job that exits with one of the specified codes is restarted.

     Specifying `REQUEUE_EXIT_VALUES=all ~0 ~1` specifies that jobs that end with an exit code other than 0 (success) or 1 (warnings) are requeued.

     Note: If you specify a return_code value that is greater than 255, LSF uses the modulus of the value with 256. For example, if SAS returns an exit code of 999, LSF sees that value as (999 mod 256) or 231. Therefore, you must specify a value of 231 on REQUEUE_EXIT_VALUES.

   - `RERUNNABLE=YES`. This specifies that jobs sent from this queue can be rerun if the host that runs them fails.
2. Specify the queue that you created in step 1, either by modifying a grid server definition or by specifying the -GRIDJOBOPTS option.

   To create or modify a grid server definition, use the Server Manager plug-in for SAS Management Console. To specify the queue, specify *"queue=name_of_requeue_queue"* in the **Additional Options** field of the server definition.

   To use -GRIDJOBOPTS, submit the job using the `-GRIDJOBOPTS queue=`*`name_of_requeue_queue`* option.
3. Submit the job to the requeue queue on the grid. You must use the SAS Grid Manager Client Utility to specify the -GRIDRESTARTOK option. Send the job to the requeue queue by using the server that you specified in step 2.

PART 6

SAS Grid Language Reference

16

SAS Functions for SAS Grid

Dictionary

GRDSVC_ENABLE Function

Enables or disables one or all SAS sessions on a grid.

Valid in: %SYSFUNC or %QSYSFUNC Macro, DATA step

Category: Grid

Syntax

grdsvc_enable(*identifier* <,*option-1*; ... *option-n*>)

grdsvc_enable(*identifier*,"" | '')

Required Argument

identifier

specifies one or all server sessions to be enabled or disabled for grid execution. The *identifier* is specified as follows:

server-ID

specifies the name of a SAS/CONNECT server session to be enabled or disabled for grid execution.

You use this *server-ID* when you sign on to a server session using the SIGNON or the RSUBMIT statement. For information about ways to specify the server ID, see *SAS/CONNECT User's Guide*.

Requirement If the function is used in a DATA step, enclose *server-ID* in double or single quotation marks. A *server-ID* cannot exceed eight characters.

ALL

specifies that all SAS sessions are enabled or disabled for grid execution.

See SIGNON statement and RSUBMIT statement in *SAS/CONNECT User's Guide*

Example

```
%let rc=%sysfunc(grdsvc_enable(grdnode1,server=SASApp));
%let rc=%sysfunc(grdsvc_enable(_all_,server=SASApp));
%let rc=%sysfunc(grdsvc_enable(notgrid1,""));
```

Optional Arguments

SASAPPSERVER=*server-value*

specifies the name of a SAS Application Server that has been defined in the SAS Metadata Repository. The SAS Application Server contains the definition for the logical grid server that defines the grid environment.

Alias SERVER=, RESOURCE=

Restriction Although a SAS Application Server is configured as a required grid resource in most environments, some grids are not partitioned by resource names. In these environments, passing the SAS Application Server name as a required resource causes the job to fail. To find out whether the SAS Application Server is designated as a required resource value or not in the SAS Metadata Repository, use the GRDSVC_GETINFO function call.

Interaction The name of the SAS Application Server is passed to Platform Suite for SAS as a resource value. When the job is executed, the grid provider selects a grid node that meets the requirements that are specified by this value. If *SAS-application-server* contains one or more spaces, the spaces are converted to underscores before the name is passed to Platform Suite for SAS as a resource value.

Tip For Platform Suite for SAS, this *server-value* corresponds with the value of a resource that the LSF administrator has configured in the lsf.cluster.*cluster-name* file and the lsf.shared file on the grid-control server.

See "GRDSVC_GETINFO Function" on page 268 to find out whether the SAS Application Server is designated as a required resource value in the SAS Metadata Repository. To remove the SAS Application Server name as a required resource, see "Modifying SAS Logical Grid Server Definitions" on page 210.

Example

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

GRIDOPTSET="*grid_options_set_name*"

specifies the grid options set to be used when running jobs on the grid. A grid options set is a collection of grid options, SAS options, and required resources that are associated with a particular SAS client application. The purpose of a grid options set is to enable a SAS grid administrator to define a collection of options in SAS metadata. These options will map to one or more SAS client applications and will be applied to workload submitted to the grid.

Requirements A grid options set must be defined in SAS Management Console before it can be used in the GRDSVC_ENABLE function.

The grid options set name must be enclosed in quotation marks.

WORKLOAD=*workload-value*

identifies the resource for the job to be executed on the grid. This value specifies an additional resource requirement for which Platform Suite for SAS selects the appropriate grid nodes.

The specified workload value should match one of the workload values that are defined in the SAS Application Server in the SAS Metadata Repository.

Requirement Workload values are case sensitive.

Interaction If *workload-value* contains one or more spaces, the spaces are converted to underscores before the value is passed to the grid provider. If *workload-value* is not located in the SAS Application Server definition and no other errors occur, a 0 result code is returned, and this note is displayed:

```
NOTE: Workload value "gridResource" does not exist in
the SAS Metadata Repository
```

.

Tip For Platform Suite for SAS, this *workload-value* corresponds with the resource that the LSF administrator has configured in the lsf.cluster.*cluster-name* file and the lsf.shared file on the grid-control computer.

Example

```
%let
rc=%sysfunc(grdsvc_enable(grdnode1, server=SASApp;
workload=EM));
```

The workload value EM specifies the resource name. EM must be assigned to a grid node in order to process this job. An example is assigning EM to machines that can process SAS Enterprise Miner jobs.

JOBOPTS=*job-opts-macro-variable*

specifies the macro variable that contains the job options. The job option name/value pairs are assigned to *job-opts-macro-variable*.

The job options are used by the grid job to control when and where a job runs. Job options are specified as name/value pairs in this format:

```
option-1=value-1;option-2="value-2 with spaces"; ...
option-n='value-n with spaces';
```

For a list of the job options that you can specify, see Appendix 2, “Supported Job Options - SAS Grid Manager for Platform,” on page 307.

Requirement Use a semicolon to separate job option and value pairs. For multiple values, use a macro quoting function for the semicolon or use single or double quotation marks to enclose all job options. If the value contains one or more spaces, tabs, semicolons, or quotation marks, enclose the value in single or double quotation marks.

Example

```
%let
rc=%sysfunc(grdsvc_enable(all, server=SASApp; jobopts=hrqueue));
%let hrqueue=queue=priority%str(;)project="HR Monthly";
signon grdnode1;
%let hrqueue='queue=priority;project="HR Yearly"';
signon grdnode2
```

Both jobs are sent to the priority queue. The first job is associated with the project named “HR Monthly” and the second job is associated with the project named “HR Yearly.”

"" | ''

disables grid execution for the specified server ID or all server sessions.

This value is intended to be used when you have specified _ALL_ in a previous call but you want to disable it for a small number of exceptions.

Requirement Double or single quotation marks can be used. Do not insert a space between the double or single quotation marks.

Interaction When quotation marks are used with _ALL_, it clears all previous grid settings that were specified using the GRDSVC_ENABLE function.

Example

```
%let rc=%sysfunc(grdsvc_enable(grdnode1,""));
%let rc=%sysfunc(grdsvc_enable(_all_,''));
```

Details

The GRDSVC_ENABLE function is used to enable and disable a grid execution. Grid execution can be enabled for a specified SAS session or for all SAS grid sessions. If a grid environment is not configured or is unavailable, the job is started as a symmetric multi-processor (SMP) process instead.

The GRDSVC_ENABLE function does not resolve to a specific grid node, and it does not cause grid execution. The server ID is mapped to a specific grid node. The server session starts on the grid node when requested by subsequent SAS statements (for example, when the SIGNON statement or the RSUBMIT statement is executed).

In order to restrict the use of specific grid nodes to be used by server sessions, the name of the SAS Application Server and the workload resource value are passed as required resources to Platform Suite for SAS.

Note: An exception to this behavior is when the SAS Application Server is disabled as a required resource for the grid server. For details, see the restriction for the SASAPPSERVER= option.

The grid can be partitioned according to resource or security requirements. If grid nodes do not have the required resources, then SAS requests fail. If grid nodes have the required resources but are busy, SAS requests are queued until grid resources become available. For information, see "Defining and Specifying Resources" on page 124.

Some SAS applications are appropriate for execution in a grid environment, but not in an SMP environment. Such applications should contain a macro that checks the return code from the GRDSVC_ENABLE function to ensure that a grid node, rather than an SMP process, is used.

Here are the result codes:

Table 16.1 *GRDSVC_ENABLE Function Result Codes*

Result Code	Explanation
2	Reports that one or all server sessions were disabled from grid execution.
1	Reports that a grid environment is unavailable due to one or more of these conditions: ■ A connection to the SAS Metadata Server is unavailable. ■ A logical grid server has not been defined in the SAS Metadata Repository. ■ The current user identity does not have authorization to use the specified logical grid server. ■ SAS Grid Manager has not been licensed. Instead, server sessions execute on the multi-processor (SMP) computers as a SASCMD sign-on. One of these commands, in order of precedence, is used to start the server session: ■ the value of the SASCMD system option ■ `!sascmd -noobjectserver`
0	Reports that the specified session was enabled.
-1	Reports a syntax error in the function call. An example is the omission of the server ID.
-2	Reports a parsing error in the function call. An example is an invalid option.

Result Code	Explanation
-3	Reports an invalid server ID in the function call.
-5	Reports an out-of-memory condition while the function is executing.
-6	Reports that the function cannot connect to the SAS Metadata Server or cannot access the grid metadata information. This condition frequently occurs when the user is not explicitly defined in metadata. By default, users without a definition in metadata are assigned to the PUBLIC group, which is not granted the ReadMetadata permission.

See Also

- *SAS/CONNECT User's Guide*
- *SAS/CONNECT User's Guide*
- *SAS Language Reference: Dictionary*
- *SAS Macro Language: Reference*

GRDSVC_GETADDR Function

Reports the IP address of the grid node on which the SAS session was chosen to execute.

Valid in: %SYSFUNC or %QSYSFUNC Macro, DATA step

Category: Grid

Syntax

grdsvc_getaddr(*identifier*)

Without Arguments

Required Argument

identifier
identifies the server session that is executing on the grid. The identifier can be specified as follows:

""
"

is an empty string that is used to refer to the computer on which the function is executing.

server-ID

specifies the server session that is executing on a grid.

You use the same *server-ID* that was used to sign on to a server session using the RSUBMIT statement or the SIGNON statement. Each server ID is associated with a fully qualified domain name (FQDN). The name resolution system that is part of the TCP/IP protocol is responsible for associating the IP address with the FQDN. The output is one or more IP addresses that are associated with the server. IP addresses are represented in IPv4 and IPv6 format, as appropriate.

Requirement Double or single quotation marks can be used. Do not insert a space between the double or single quotation marks.

Interaction If the function is used in a DATA step, enclose *server-ID* in double or single quotation marks.

Example

```
/*-------------------------------------------------------------------
*/
/* The following sets the macro variable 'myip' to the IP address
*/
/* of the grid node associated with the server session 'task1'
*/
/*-----------------------------------------------------------------
*/
  %let
myip=%sysfunc(grdsvc_getaddr(task1));
```

See Also

RSUBMIT statement

- *SAS/CONNECT User's Guide*

SIGNON statement

- *SAS/CONNECT User's Guide*

DATA step

- *SAS Language Reference: Dictionary*

%SYSFUNC or %QSYSFUNC

- *SAS Macro Language: Reference*

GRDSVC_GETINFO Function

Reports information about the grid environment.

Valid in: %SYSFUNC or %QSYSFUNC Macro, DATA step

Category: Grid

Syntax

grdsvc_getinfo(*identifier*)

Required Argument

identifier

specifies the server session or the SAS Application Server whose details you want to have reported to the SAS log.

The *identifier* is specified as follows:

server-ID

reports details about the specified server ID. The details that are returned by the GRDSRV_INFO function reflect the arguments that are specified in the GRDSVC_ENABLE function. You can request details about a `server-ID` that you have used to create a server session or that you will use to create a server session on the grid.

Requirement A *server-ID* cannot exceed eight characters.

ALL

reports details about all server IDs to the SAS log. The details that are returned by the GRDSRV_INFO function reflect the arguments that are specified in the GRDSVC_ENABLE function.

SASAPPSERVER=*SAS-application-server*

reports information about the specified SAS Application Server to the SAS log.

Alias SERVER=, RESOURCE=

SHOWID

lists each server session and its status: enabled for grid execution, enabled for SMP execution, or disabled.

Interaction If the GRDSVC_GETINFO function is used in a DATA step, enclose the *identifier* in single or double quotation marks. The identifier can be specified as *server-ID*, _ALL_, SASAPPSERVER=*SAS-application-server*, or _SHOWID_. If no grid processes were enabled using the GRDSRV_ENABLE function or if all grid processes were disabled using the GRDSVC_ENABLE function with _ALL_ option, this message is displayed:

```
NOTE: No remote session ID enabled/disabled for the grid service.
```

Tip You do not have to be signed on to a specific server session in order to get information about it.

Example This log message reports that the SAS Application Server is a required resource.

```
%put
%sysfunc(grdsvc_getinfo(server=SASApp));
NOTE: SAS Application Server Name= SASAPP
            Grid Provider=   Platform
            Grid Workload=   gridwrk
            Grid SAS Command=   gridsasgrid
            Grid Options=   gridopts
            Grid Server Addr=   d15003.na.sas.com
            Grid Server Port=   123
            Grid Module=   gridmod
            Server name is a required grid resource value.
```

If the SAS Application Server is a disabled required resource, this message is displayed:

```
Server name is not a required grid resource value.
```

Optional Arguments

CLIENTAPP="*client_application*"
returns information about the grid options set associated with the specified application. The information returned includes the grid options, SAS options, and required resources that are specified in the grid options set and are associated with the specified application.

GRIDOPTSET="*grid_options_set_name*"
returns information about the specified grid options set. A grid options set is a collection of grid options, SAS options, and required resources that are associated with a particular SAS client application. The purpose of a grid options set is to enable a SAS grid administrator to define a collection of options in SAS metadata. These options will map to one or more SAS client applications and will be applied to workload submitted to the grid.

Requirement The grid options set name must be enclosed in quotation marks.

Example This example returns information about the grid options set "OptionsSet1."

```
%put %sysfunc(grdsvc_getinfo(server=CNTWin;
gridoptset="OptionsSet1"));
NOTE: SAS Application Server Name= CNTWin
      Grid Provider=               Platform
      Grid Workload=               idbvm1, idbvm2, idbvm3
      Grid SAS Command=            gridsas
      Grid Server Addr=            rlsgrid.mycorp.com
      Grid Required Resources=     <none>
      Grid SAS options=            -set GAOvar OptionsSet1
```

Details

Here are the result codes:

Table 16.2 GRDSVC_GETINFO Function Return Codes

Result Code	Explanation
2	Reports that the specified server ID is not enabled for grid execution.
1	Reports that the specified server ID is enabled for SMP execution.
0	Reports that the specified server ID is enabled for a grid execution or that no error occurred.
-1	Reports a syntax error in the function call. An example is that an empty string is specified for the server ID.
-2	Reports a parsing error in the function call. An example is the failure to specify the SAS Application Server using the SASAPPSERVER= option.
-3	Reports an invalid server ID in the function call.
-5	Reports an out-of-memory condition while the function is executing.
-6	Reports that an error occurred when the SAS Metadata Server was accessed or when the information was returned from the SAS Metadata Server.

Example

```
/
*-----------------------------------------------------------------
--*/
/* Show grid logical server definition for SAS Application Server
'SASApp'*/
/
*-----------------------------------------------------------------
--*/
%let rc=%sysfunc(grdsvc_getinfo(sasappserver=SASApp));
/
*-----------------------------------------------------------------
--*/
/* Show grid information about server session ID
'task1'                  */
```

```
/
*-----------------------------------------------------------------
--*/
%let rc=%sysfunc(grdsvc_getinfo(task1));
/
*-----------------------------------------------------------------
--*/
/* Show server session information for all server
sessions                */
/
*-----------------------------------------------------------------
--*/
%let rc=%sysfunc(grdsvc_getinfo(_ALL_));
/
*-----------------------------------------------------------------
--*/
/* Show all server session IDs that are either grid-enabled
or            */
/* grid-
disabled                                                        */
/
*-----------------------------------------------------------------
--*/
%let
rc=%sysfunc(grdsvc_getinfo(_SHOWID_));
```

See Also

RSUBMIT statement

- *SAS/CONNECT User's Guide*

SIGNON statement

- *SAS/CONNECT User's Guide*

DATA step

- *SAS Language Reference: Dictionary*

%SYSFUNC or %QSYSFUNC

- *SAS Macro Language: Reference*

GRDSVC_GETNAME Function

Reports the name of the grid node on which the SAS grid server session was chosen to execute.

Valid in: %SYSFUNC or %QSYSFUNC Macro, DATA step

Category: Grid

Syntax

grdsvc_getname(*identifier*)

Required Argument

identifier

identifies the server session that is executing on the grid. The identifier can be specified as follows:

""
"

is an empty string that is used to refer to the computer at which the statement is executed.

server-ID

specifies the server session that is executing on a grid.

You use the same *server-ID* that you used to sign on to a server session using the RSUBMIT statement or the SIGNON statement .

If the function is used in a DATA step, enclose *server-ID* in double or single quotation marks.

Example

```
/
*-----------------------------------------------------------------
-*/
/* The following sets the macro variable 'mynodea' to the name
of        */
/* the grid node associated with the server ID
'task1'.                  */
/
*-----------------------------------------------------------------
-*/
%let
mynodea=%sysfunc(grdsvc_getname(task1));
```

See Also

RSUBMIT statement

- *SAS/CONNECT User's Guide*

SIGNON statement

- *SAS/CONNECT User's Guide*

DATA step

- *SAS Language Reference: Dictionary*

%SYSFUNC or %QSYSFUNC

- *SAS Macro Language: Reference*

GRDSVC_HOSTLIST Function

Returns a list of hosts that are available to run grid jobs. The list is ordered according to the priority set by the grid provider.

Valid in: %SYSFUNC or %QSYSFUNC Macro, DATA step

Category: Grid

Syntax

grdsvc_hostlist(‘SERVER=*SAS-application-server*;
HOSTLIST=*grid_host_machine,[grid_host_machine]* |
WORKLOAD=*workload-value* ||
CLIENTAPP=”*client_application*” |
GRIDOPTSET=”*grid_options_set_name*”;’)

Required Argument

SASAPPSERVER=*SAS-application-server*
specifies the name of the SAS Application Server that has been defined in the SAS Metadata Repository. The SAS Application Server contains the definition for the logical grid server that is used to access the grid environment. The name of the SAS Application Server is passed to the grid provider as a required resource.

Optional Arguments

HOSTLIST=*grid_host_machine,[grid_host_machine]*
specifies a comma-separated list of host machines that are on the grid and might be available to process grid jobs. If the list includes machines that are not included in the grid definition or are not available to process jobs, they are not included in the output of the function. If none of the machines in the list are included on the grid, the function returns an error message.

WORKLOAD=*workload-value*
identifies the resource for the job to be executed on the grid. This value specifies an additional resource requirement for which Platform Suite for SAS selects the appropriate grid nodes.

The specified workload value should match one of the workload values that are defined in the SAS Application Server in the SAS Metadata Repository.

Requirement Workload values are case sensitive.

Interaction If *workload-value* contains one or more spaces, the spaces are converted to underscores before the value is passed to the grid provider. If *workload-value* is not located in the SAS Application Server definition and no other errors occur, a 0 result code is returned, and this note is displayed:

```
NOTE: Workload value "gridResource" does not exist in
the SAS Metadata Repository
```

.

Tip For Platform Suite for SAS, this *workload-value* corresponds with the resource that the LSF administrator has configured in the lsf.cluster.*cluster-name* file and the lsf.shared file on the grid-control computer.

Example

```
%let
rc=%sysfunc(grdsvc_enable(grdnode1, server=SASApp;
workload=EM));
```

The workload value EM specifies the resource name. EM must be assigned to a grid node in order to process this job. An example is assigning EM to machines that can process SAS Enterprise Miner jobs.

CLIENTAPP="*client_application*"
returns information about the hosts that are available to run jobs from the specified application.

GRIDOPTSET="*grid_options_set_name*"
returns information about the hosts associated with the specified grid options set. A grid options set is a collection of grid options, SAS options, and required resources that are associated with a particular SAS client application. The purpose of a grid options set is to enable a SAS grid administrator to define a collection of options in SAS metadata. These options will map to one or more SAS client applications and will be applied to workload submitted to the grid.

Requirement The grid options set name must be enclosed in quotation marks.

Example

The code

```
%put
%sysfunc(grdsvc_hostlist('server=SASApp;hostlist=node1,node2,node3;'));
```

returns the output

```
node1,node3
```

This output indicates that node1 and node3 machines are available to process grid jobs, and that node1 is first priority to process jobs.

GRDSVC_NNODES Function

Reports the total number of job slots that are available for use on a grid.

Valid in: %SYSFUNC or %QSYSFUNC Macro, DATA step

Category: Grid

Syntax

grdsvc_nnodes(*argument*;*option*)

Required Argument

SASAPPSERVER=*SAS-application-server*

specifies the name of the SAS Application Server that has been defined in the SAS Metadata Repository. The SAS Application Server contains the definition for the logical grid server that is used to access the grid environment. The name of the SAS Application Server is passed to Platform Suite for SAS as a required resource. The grid provider selects the grid nodes that meet the requirements for the specified SAS Application Server and returns the total number of job slots on the grid.

An exception to this behavior is when the SAS Application Server is disabled as a required resource for the grid server. For details, see the SASAPPSERVER= option for the GRDSVC_ENABLE function on page 261.

Alias SERVER=, RESOURCE=

Interaction If *SAS-application-server* contains one or more spaces, the spaces are converted to underscores before the name is passed to Platform Suite for SAS .

Example

```
%let
numofnodes%sysfunc(grdsvc_nnodes(server=SASApp));
```

Optional Argument

WORKLOAD=*workload-value*

identifies the resource for the type of job to be executed on the grid. This value specifies the workload requirements for which Platform Suite for SAS selects the grid nodes that contain these resources.

The specified workload value should match one of the workload values that are defined in the SAS Application Server in the SAS Metadata Repository.

Requirement If you specify WORKLOAD=, you must also specify the SASAPPSERVER= option. Workload values are case sensitive.

Interaction If *workload-value* contains one or more spaces, the spaces are converted to underscores before the value is passed to Platform Suite for SAS . If *workload-value* is not located in the SAS Application Server definition and no other errors occur, a 0 result code is returned. A 0 result code means that no grid nodes contain the requested resources. Also, this note is displayed:

```
NOTE: Workload value "gridResource" does not exist in
the SAS Metadata Repository.
```

If *workload-value* is undefined to Platform Suite for SAS , the GRDSVC_NNODES function returns the result code 0.

Tip For Platform Suite for SAS, this *workload-value* corresponds with the resource that the LSF administrator has configured in the lsf.cluster.*cluster-name* file and the lsf.shared file on the grid-control computer.

Example

```
%let
numofnodes=%sysfunc(grdsvc_nnodes(server=SASApp; workload=em));
```

The workload value, EM , specifies the resource name. EM must be assigned to a grid node in order to process this job. An example is assigning EM to machines that can process SAS Enterprise Miner jobs.

Details

When a grid environment is available, the GRDSVC_NNODES function returns the total number of job slots (busy and idle) that are available for job execution. This value is resolved when the function is called. Because of this, the value might vary over time, according to whether job slots have been added or removed from the grid. It can also vary based on the user, the queue that is being used, or other slot limits that are defined in the LSF configuration.

Note: If you are using SAS Grid Manager for Hadoop, the value returned by the GRDSVC_NNODES function might not match the number of jobs that can actually be executed at the same time. In a SAS Grid Manager for Hadoop environment, jobs contain resources for the app master as well as resources for the actual job. Because the app master and the job will sometimes run on the same machine and will sometimes run on different machines, an exact count of the number of executing jobs cannot be determined.

Here are the result codes:

Table 16.3 *GRDSVC_NNODES Function Result Codes*

Result Code	Explanation
nnn	If a grid environment is available, reports the total number of job slots (idle and busy) that have been configured in a grid environment. The grid contains the resources that are specified by the SASAPPSERVER= argument and the WORKLOAD= option.

Result Code	Explanation
	If a grid environment is not available, assumes a multi-processor (SMP) environment, and reports the value of the CPUCOUNT system option. In this case, the lowest value that can be reported is 1.
1	If a grid environment is not available, assumes a multi-processor (SMP) environment, and reports the value of the CPUCOUNT system option. In this case, the lowest value that can be reported is 1.
0	Reports that no grid nodes contain the requested resources.
-1	Reports a syntax error in the function call. For example, a syntax error would result from supplying no value, or an empty string, to the SASAPPSERVER= option.

Example

```
/
*-----------------------------------------------------------------------
-*/
/* Get the number of grid nodes that have 'SASApp' as a
resource         */
/
*-----------------------------------------------------------------------
-*/
%let NumNodes=%sysfunc(grdsvc_nnodes(server=SASApp));
/
*-----------------------------------------------------------------------
-*/
/* Get the number of grid nodes that have 'SASApp'  'EM' as
resources    */
/
*-----------------------------------------------------------------------
-*/
%let
numofnodes=%sysfunc(grdsvc_nnodes(server=SASApp;workload=EM));
```

See Also

RSUBMIT statement

- *SAS/CONNECT User's Guide*

SIGNON statement

- *SAS/CONNECT User's Guide*

DATA step

- *SAS Language Reference: Dictionary*

CPUCOUNT= system option

- *SAS Language Reference: Dictionary*

GRDSVC_OPTSETS Function

Returns a list of valid options sets for a specified combination of SAS Application Server, SAS client application, and user. If the user is not specified, the function returns a list of all valid grid options sets that apply to the specified SAS client application.

Valid in: %SYSFUNC or %QSYSFUNC Macro, DATA step

Category: Grid

Syntax

grdsvc_optsets(“SERVER=*SAS-application-server*;CLIENTAPP=’*SAS-client-application*’;
USER=’*user-ID*’;”)

Required Arguments

SERVER=*SAS-application-server*
specifies the name of the SAS Application Server that has been defined in the SAS Metadata Repository. The SAS Application Server contains the definition for the logical grid server that is used to access the grid environment. The name of the SAS Application Server is passed to the grid provider as a required resource.

CLIENTAPP=’*SAS-client-application*’
specifies the name of a SAS client application for which a grid options set has been created.

Optional Argument

USER=’*user-ID*’
specifies the ID of a user defined in metadata who might be subject to a grid options set for the specified SAS client application. The grid options set can apply to the user explicitly or as a member of a user group.

Example

The code

```
%put
```

```
%sysfunc(grdsvc_optsets("server=GOSWin;clientapp='SAS Grid Manager
Client
Utility';user='sasuser1';"));
```

returns the output

```
"Sasuser1_optset_SASSGSUB","Group1_optset_SASGSUB"
```

which are the two grid options sets that apply to user sasuser1 for the SAS Grid Manager Client Utility. The options set Sasuser_optset_SASGSUB is explicitly defined for user sasuser1. The options set Group1_optset_SASGSUB applies because the user belongs to group Group1.

17

SASGSUB Command

SASGSUB Overview

SAS Grid Manager Client Utility is a command-line utility that enables users to submit SAS programs, operating system commands, or command files to a grid for processing and to start interactive sessions on grid machines. This utility allows a grid client to submit SAS programs to a grid without requiring that SAS be installed on the machine performing the submission. It also enables jobs to be processed on the grid without requiring that the client remain active.

You can use the SAS Grid Manager Client Utility's SASGSUB command to submit jobs to the grid, start interactive sessions, view job status, retrieve results, and terminate jobs. The SAS Grid Manager Client Utility options can be specified in a configuration file so that they do not have to be entered manually. SASGSUB uses the sasgsub.cfg configuration file, which contains the required options. This file is automatically created by the SAS Deployment Wizard during installation. It stores the file in ***config_dir***`/Applications/SASGridManagerClientUtility/`***version***.

If you are using SAS Grid Manager for Platform, Platform LSF must be installed on any machine where the SAS Grid Manager Client Utility runs.

Disabling SASGSUB Actions

You might want to prevent users from using the SAS Grid Manager Client Utility to perform certain actions, such as terminating jobs. To disable these actions, you can specify these SasgsubDisable extended attributes on the grid server metadata definitions:

SasgsubDisableGRIDSUBMITPGM
: prevents SAS jobs from being submitted

SasgsubDisableGRIDKILLJOB
: prevents jobs from being terminated

SasgsubDisableGRIDRUNCMD
: prevents arbitrary commands from running in batch

SasgsubDisableGRIDRUNCMDINT
: prevents arbitrary commands from running in an interactive session

SasgsubDisableGRIDRUNSASLM
: prevents an interactive SAS line mode session from starting

SasgsubDisableGRIDRUNSASDMS
: prevents an interactive SAS Display Manager session from starting

You do not need to specify a value for these extended attributes.

Dictionary

SASGSUB Command: Submitting a SAS Program in Batch Mode

This is the complete syntax for submitting a SAS program to a grid in batch mode. Enter the command on a Windows or UNIX command line.

Syntax

SASGSUB
: -GRIDSUBMITPGM *sas-program-file*
  common_arguments batch_arguments

<-GRIDRESTARTOK | -GRIDLRESTARTOK>
<-GRIDSASOPTS *grid-sas-options*>
<*grid_job_arguments*>

Required Arguments

-GRIDSUBMITPGM *sas-program-file*
specifies the path and filename of the SAS program that you want to run on the grid.

Note: The path cannot contain spaces, tabs, or any of these characters:

> < & | ^ , ; =

Alias SUBMITPGM

common_arguments
specifies any of the arguments that are common to all uses of the SASGSUB command. Some arguments are required and some arguments are optional. See "SASGSUB Command: Common Arguments" on page 293 for the syntax for each argument.

batch_arguments
specifies any of the arguments that are common to all SASGSUB commands that run in batch mode. Some arguments are required and some are optional. See "SASGSUB Command: Batch Options" on page 298 for the syntax for each argument.

Optional Arguments

-GRIDRESTARTOK | -GRIDLRESTARTOK
specifies that the job can be restarted at a checkpoint or a label.

Alias (for GRIDRESTARTOK) RESTARTOK, RESTART

Alias (for GRIDLRESTARTOK) LRESTARTOK, LRESTART

-GRIDSASOPTS *grid-sas-options*
specifies any SAS options that are applied to the SAS session started on the grid. Multiple options can be enclosed in quotation marks. If you are specifying options that contain quotation marks, you can enclose all of the options in parentheses before enclosing them in quotation marks. Each of these forms is valid:

GRIDSASOPTS *option*

GRIDSASOPTS "*option1 option2*"

GRIDSASOPTS "(*option1 option2 option3*)"

GRIDSASOPTS "(*option1 option2* '*option3*')"

When SASGSUB runs, the outer set of quotation marks is removed.

Alias SASOPTS

grid_job_arguments
specifies any of the arguments valid when submitting jobs or commands to the grid. See "SASGSUB Command: Grid Job Arguments" on page 300 for the syntax for each of these arguments.

Example: Submitting a SAS Program to the Grid in Batch Mode

This is an example of a SASGSUB statement used to submit the SAS program Lab_report.sas in the directory **C:\SAS_programs** to the grid in batch mode. The job has been enabled for restarting, and it needs to run in the overnight queue. The grid uses a shared file system.

```
SASGSUB -GRIDSUBMITPGM C:\SAS_programs\Lab_report.sas -RESTARTOK
-JOBOPTS queue=overnight
```

SASGSUB Command: Submitting a SAS Program Located on the Grid

This is the complete syntax for submitting a SAS program that exists on a shared filesystem and that is therefore shared across all grid nodes, rather than on an internal filesystem and accessible only to one host. Because the program is already on a shared filesystem, it does not have to be copied to a grid share when it runs. The log and output of the program are saved to the job directory. Enter the command on a Windows or UNIX command line.

Syntax

SASGSUB
-GRIDRUNPGM *path filename*

Required Argument

-GRIDRUNPGM *path filename*
specifies the path and filename of a SAS program that you want to run on the grid. The program is stored on a grid node machine, and is not accessible to users on a client machine. The *path* and *filename* values must be same as specified on the grid.

Note: The *path* cannot contain spaces, tabs, or any of these characters:

> < & | ^ , ; =

Alias RUNPGM

SASGSUB Command: Starting SAS in Interactive Mode

This is the complete syntax for starting SAS in interactive mode. You can use either line mode or SAS Display Manager mode to run SAS in a UNIX grid, with input coming from and output going to the client machine. SAS Display Manager mode also requires that an X server be present on the client machine. If you terminate SASGSUB in interactive mode, you also terminate the SAS session running on the grid.

Syntax

SASGSUB
  -GRIDRUNSASLM | -GRIDRUNSASDMS
    <hostname:>displaynumber<.screennumber>
    common_arguments
    <-GRIDWAITTIMEOUT *seconds*>
    <grid_job_arguments>

Required Arguments

-GRIDRUNSASLM
specifies that the SASGSUB command runs a SAS session in line mode. Standard output and standard error are displayed by SASGSUB. SASGSUB also redirects standard input to the SAS session running on the grid.

Alias RUNSASLM

-GRIDRUNSASDMS *<hostname:>displaynumber<.screennumber>*
specifies that the SASGSUB command runs in SAS Display Manager mode. Standard output and standard error are displayed by SASGSUB. The X Windows session is directed to the host and port specified by the *hostname:displaynumber.screennumber* options.

Alias RUNSASDMS

common_arguments
specifies any of the arguments that are common to all uses of the SASGSUB command. Some arguments are required and some are optional. See "SASGSUB Command: Common Arguments" on page 293 for the syntax for each argument.

Optional Arguments

-GRIDWAITTIMEOUT *timeout-value*
specifies that the interactive SAS session is terminated when the *timeout-value* (in seconds) is reached.

Alias WAITTIMEOUT, TIMEOUT

grid_job_arguments
specifies any of the arguments that are valid when submitting jobs or commands to the grid. See "SASGSUB Command: Grid Job Arguments" on page 300 for the syntax for each of these arguments.

SASGSUB Command: Running a Command in Batch Mode

This is the complete syntax for submitting a grid job to run a command in batch mode.

Syntax

SASGSUB
-GRIDRUNCMD *command common_arguments batch_arguments <grid_job_arguments>*

Required Arguments

-GRIDRUNCMD *command*
specifies a command that you want to run on the grid.

Alias RUNCMD

common_arguments
specifies any of the arguments that are common to all uses of the SASGSUB command. Some arguments are required and some are optional. See "SASGSUB Command: Common Arguments" on page 293 for the syntax for each argument.

batch_arguments
specifies any of the arguments that are common to all SASGSUB commands that run in batch mode. Some arguments are required and some are optional. See "SASGSUB Command: Batch Options" on page 298 for the syntax for each argument.

Optional Argument

grid_job_arguments
specifies any of the arguments valid when submitting jobs or commands to the grid. See "SASGSUB Command: Grid Job Arguments" on page 300 for the syntax for each of these arguments.

Example: Submitting a Command in Batch Mode

This is an example of a SASGSUB batch mode statement used to submit a command to the grid to copy a set of files from the **prod** directory to the **backup** directory:

```
SASGSUB -GRIDRUNCMD "cp /prod/file1.* /backup"
```

SASGSUB Command: Running a Command in Interactive Mode

This is the complete syntax for submitting a grid job to run a command interactively. If you terminate SASGSUB in interactive mode, you also terminate the SAS session running on the grid.

Syntax

SASGSUB
-GRIDRUNCMDINT *command common_arguments*
<-GRIDWAITTIMEOUT>*timeout-value*
<*grid_job_arguments*>

Required Arguments

-GRIDRUNCMDINT *command*
specifies that an arbitrary command is run on the grid in interactive mode. Standard output and standard error is displayed on the user's machine. Standard input is redirected from the client to the grid.

Alias RUNCMDINT

common_arguments
specifies any of the arguments that are common to all uses of the SASGSUB command. Some arguments are required and some are optional. See "SASGSUB Command: Common Arguments" on page 293 for the syntax for each argument.

Optional Arguments

-GRIDWAITTIMEOUT *timeout-value*
specifies that the SAS Grid Manager Client Utility waits until either the job has completed running (either successfully or with an error) or *timeout-value* (in seconds) is reached. If you use this option with -WAIT or -WAITRESULTS, it enables you to specify how long to wait for results.

Alias WAITTIMEOUT, TIMEOUT

grid_job_arguments
specifies any of the arguments that are valid when submitting jobs or commands to the grid. See "SASGSUB Command: Grid Job Arguments" on page 300 for the syntax for each of these arguments.

SASGSUB Command: Ending a Job

This is the complete syntax for ending a job on a SAS grid. Enter the command on a Windows or UNIX command line.

Syntax

SASGSUB
-GRIDKILLJOB *job-id* | ALL *common_arguments*

Required Arguments

-GRIDKILLJOB *job-id* | ALL
terminates the job specified by *job-id*. If you specify ALL, all jobs are terminated.

Alias KILLJOB

common_arguments
specifies any of the arguments that are common to all uses of the SASGSUB command. Some arguments are required and some are optional. See "SASGSUB Command: Common Arguments" on page 293 for the syntax for each argument.

Example: Ending a Job

This is an example of a SASGSUB statement used to end the job 61361 that is running on the grid:

```
SASGSUB -GRIDKILLJOB 61361
```

SASGSUB Command: Viewing Job Status

This is the syntax for using SASGSUB to view the status of a job on a SAS grid. Enter the command on a Windows or UNIX command line.

Syntax

SASGSUB
-GRIDGETSTATUS *job-id* | ALL
common_arguments batch_arguments

Required Arguments

-GRIDGETSTATUS *job-id* | ALL
displays the status of the job specified by *job-id*. If you specify ALL, the status of all jobs for the current user is displayed.

Alias GETSTATUS

common_arguments
specifies any of the arguments that are common to all uses of the SASGSUB command. Some arguments are required and some are optional. See "SASGSUB Command: Common Arguments" on page 293 for the syntax for each argument.

batch_arguments
specifies any of the arguments that are common to all SASGSUB commands that run in batch mode. Some arguments are required and some are optional. See "SASGSUB Command: Batch Options" on page 298 for the syntax for each argument.

Examples

Example 1: Viewing the Status of a Single Job

This is an example of a SASGSUB statement used to view the status of job 61361 that is running on the grid:

```
SASGSUB -GRIDGETSTATUS 61361
```

The output from the command looks like this:

Output 17.1 *Status of a Single Job*

```
Current Job Information
  Job 61361 (testPgm) is Finished:  Submitted: 06Jan2011:10:28:57, Started:
06Jan2011:10:28:57 on Host d12345, Ended: 06Jan2011:10:28:57
```

Example 2: Viewing the Status of All Jobs

This is an example of a SASGSUB statement used to view the status of all jobs running on the grid:

```
SASGSUB -GRIDGETSTATUS ALL
```

The output from the command looks like this:

Output 17.2 *Status of All Jobs*

```
Current Job Information
  Job 1917 (Pgm_01) is Finished:  Submitted: 08May2011:10:28:57, Started:
08May2011:10:28:57 on Host d12345, Ended: 08May2011:10:28:57
  Job 1918 (Pgm_02) is Finished:  Submitted: 08May2011:10:28:57, Started:
08May2011:10:28:57 on Host d12345, Ended: 08May2011:10:28:57
  Job 1919 (Pgm_03) is Finished:  Submitted: 08May2011:10:28:57, Started:
08Ma2011:10:28:57 on Host d12345, Ended: 08May2011:10:28:57
  Job information in directory U:\jobs\GridSub\GridWork
\user1\SASGSUB-2011-05-11_13.17.17.327_testPgm is invalid.
  Job 1925 (Pgm_04) is Submitted: Submitted: 08May2011:10:28:57
```

SASGSUB Command: Retrieving Job Output

This is the syntax for using SASGSUB to retrieve the output of a job that has completed processing on a SAS grid. Enter the command on a Windows or UNIX command line.

Syntax

SASGSUB
-GRIDGETRESULTS *job-id* | ALL
common_arguments batch_arguments
<-GRIDRESULTSDIR *directory*>
< -GRIDFORCECLEAN > <-VERBOSE>

Required Arguments

-GRIDGETRESULTS *job-id* | ALL
Copies the job information from the work directory to the directory specified by -RESULTSDIR for the specified *job-id* or for all jobs. If you do not specify a value for -RESULTSDIR, the job information is copied to the current directory. An error results if the current directory cannot be written to.

Alias GETRESULTS

common_arguments
specifies any of the arguments that are common to all uses of the SASGSUB command. Some arguments are required and some are optional. See "SASGSUB Command: Common Arguments" on page 293 for the syntax for each argument.

batch_arguments
specifies any of the arguments that are common to all SASGSUB commands that run in batch mode. Some arguments are required and some are optional. See "SASGSUB Command: Batch Options" on page 298 for the syntax for each argument.

Optional Arguments

-GRIDRESULTSDIR *directory*
specifies the directory to which the job results are moved. If you do not specify a value for this argument, the default value is used, which is the current directory.

Alias RESULTSDIR

-GRIDFORCECLEAN
specifies that the job directory on the grid is deleted, regardless of whether the job was successful or not.

Alias FORCECLEAN

Examples

Example 1: Retrieving the Output of a Job

This is an example of a SASGSUB statement used to view the output of job 61361 that is running on the grid:

```
SASGSUB -GRIDGETRESULTS 61361
```

The output from the command looks like this:

Output 17.3 *Output of a Single Job*

```
Current Job Information
  Job 61361 (testPgm) is Finished:  Submitted: 06Jan2011:10:53:33, Started:
06Jan2011:10:53:33 on Host d15003, Ended: 06Jan2011:10:53:33
    Moved job information to .\SASGSUB-2011-01-06_21.52.57.130_testPgm
```

Example 2: Retrieving the Output of All Jobs

This is an example of a SASGSUB statement used to view the output of all jobs on the grid:

```
SASGSUB -GRIDGETRESULTS ALL
```

The output from the command looks like this:

Output 17.4 *Output of All Jobs*

```
Current Job Information
  Job 1917 (Pgm1) is Finished:  Submitted: 08Dec2008:10:53:33, Started:
08May2011:10:53:33 on Host d15003, Ended: 08May2011:10:53:33
    Moved job information to .\SASGSUB-2011-05-06_21.52.57.130_Pgm1

  Job 1918 (Pgm2) is Finished:  Submitted: 08May2011:10:53:33, Started:
08May2011:10:53:33 on Host d15003, Ended: 08May2011:10:53:33
    Moved job information to .\SASGSUB-2011-05-06_13.13.39.167_Pgm2

  Job 1919 (Pgm3) is Finished:  Submitted: 08May2011:10:53:34, Started:
08May2011:10:53:34 on Host d15003, Ended: 08May2011:10:53:34
    Moved job information to .\SASGSUB-2011-05-06_13.16.06.060_Pgm3

  Job information in directory U:\jobs\GridSub\GridWork
\user1\SASGSUB-2011-05-06_13.17.17.327_testPgm is invalid.
    Moved job information to .\SASGSUB-2011-05-06_13.17.17.327_Pgm4

  Job 1925 (Pgm4) is Submitted: Submitted: 08May2011:10:53:34
```

SASGSUB, Commands: Purging Job Directories

This is the syntax for using SASGSUB to purge the job directories in the GRIDWORK location that are older than the specified age. Enter the command on a Windows or UNIX command line.

Syntax

SASGSUB
  -GRIDCLEANGRIDWORK *age-in-hours*
    -GRIDWORK *work-directory*
    common_arguments

Required Arguments

-GRIDCLEANGRIDWORK *age-in-hours*
purges the job directories in the GRIDWORK location that are older than the age specified by *age-in-hours*.

Alias CLEANGRIDWORK

-GRIDWORK *work-directory*
specifies the path for the shared directory that the job uses to store the program, output, and job information. On Windows, the path can contain spaces, but must be in quotation marks. On UNIX, the path cannot contain spaces. This option is stored in the configuration file that is automatically created by the SAS Deployment Wizard.

common_arguments
specifies any of the arguments that are common to all uses of the SASGSUB command. Some arguments are required and some are optional. See

"SASGSUB Command: Common Arguments" on page 293 for the syntax for each argument.

Example: Purging Job Directories

This is an example of a SASGSUB statement used to purge any directories in the GRIDWORK location that are age 72 hours or older:

```
SASGSUB -GRIDCLEANGRIDWORK 72 -GRIDWORK /path/to/grid/work
```

SASGSUB Command: Common Arguments

These arguments are valid for all SASGSUB commands. Argument values that contain spaces must be contained in quotation marks.

Syntax

-METASERVER *server*
-METAUSER *user-ID*
-METAPORT *port*
-METAPASS *password*
-METAPROFILE *profile-name*
-METACONNECT *connection-name*
-XMLCONFIGFILE *pathname*
-GRIDAPPSERVER *sas-application-server*
<-GRIDCONFIG *grid-option-file* >
< -GRIDLICENSEFILE *grid-enabled-license-file*>
<-LOGCONFIGLOC *logging-option-file*>
<-GRIDLIBPATH *path*>
< -GRIDSETPERMS >
<-VERBOSE>
<-GRIDUSER *grid-logon-username* | _PROMPT_>
<–GRIDPASS *grid-logon-password* | _PROMPT_>
<-GRIDPLUGINPATH *grid-jar-file-path*>
<-JREOPTIONS *java-runtime-options*>
<-SSLCALISTLOC *filename* >
<-SSLCERTISS *issuer*>
<-SSLCERTLOC *filename*>
<-SSLCERTSERIAL *serial_number*>
<-SSLCERTSUBJ *subject*>
<-SSLCLIENTAUTH >
<-SSLCRLCHECK >
<-SSLCRLLOC *filename*>
<-SSLPKCS12LOC *filename*>
<-SSLPKCS12PASS *password*>
<-SSLPVTKEYLOC *filename*>
<-SSLPVTKEYPASS *password*>

Required Arguments

-GRIDAPPSERVER *sas-application-server*
specifies the name of the SAS Application Server that contains the grid's logical grid server definition. This option is stored in the configuration file that is automatically created by the SAS Deployment Wizard.

Alias APPSERVER, SERVER

-METASERVER *server*
specifies the name or IP address of the SAS Metadata Server.

You must specify either -METASERVER, -METAPORT, -METAUSER, and -METAPASS; or -METAPROFILE and -METACONNECT; or -XMLCONFIGFILE. You cannot specify more than one group of options.

This option is stored in the configuration file that is automatically created by the SAS Deployment Wizard.

-METAPORT *port*
specifies the port to use to connect to the SAS Metadata Server specified by the -METASERVER argument. This option is stored in the configuration file that is automatically created by the SAS Deployment Wizard.

-METAUSER *user-ID*
specifies the user ID to use to connect to the SAS Metadata Server specified by the -METASERVER argument. This option is stored in the configuration file that is automatically created by the SAS Deployment Wizard. Specify _PROMPT_ for *user-ID* to specify that the SAS Grid Manager Client Utility prompts the user for a user ID.

-METAPASS *password* | _PROMPT_
specifies the password of the user specified in the -METAUSER argument. If the value of the argument is set to _PROMPT_, the user is prompted for a password. This option is stored in the configuration file that is automatically created by the SAS Deployment Wizard.

-METAPROFILE *profile_pathname*
specifies the pathname of the connection profile for the SAS Metadata Server.

You must specify either -METASERVER, -METAPORT, -METAUSER, and -METAPASS; or -METAPROFILE and -METACONNECT; or -XMLCONFIGFILE. You cannot specify more than one group of options.

This option is stored in the configuration file that is automatically created by the SAS Deployment Wizard.

If you are using a clustered metadata server, you can use -METAPROFILE to specify the clustered metadata server definition XML file. In this scenario, you do not need to specify the -METACONNECT option.

-METACONNECT *connection-name*
specifies the name of the connection to use when connecting to the SAS Metadata Server. The connection must be defined in the metadata profile specified in the -METAPROFILE argument. This option is stored in the configuration file that is automatically created by the SAS Deployment Wizard.

-XMLCONFIGFILE *pathname*
specifies the path to an XML file that contains the metadata profile connection information.

You must specify either -METASERVER, -METAPORT, -METAUSER, and -METAPASS; or -METAPROFILE and -METACONNECT; or -XMLCONFIGFILE. You cannot specify more than one group of options.

Optional Arguments

-GRIDLICENSEFILE *grid-enabled-license-file*
specifies the path and filename of a SAS license file that contains the SAS Grid Manager license. The default value is to retrieve the license file information from metadata. If specified, the location must point to a valid SID file that contains a SAS Grid Manager license. Do not use this option unless instructed by SAS Technical Support.

Alias LICENSEFILE, LICENSE

-GRIDCONFIG *grid-option-file*
specifies the path and filename of a file containing other SASGSUB options. The default value is sasgsub.cfg.

-LOGCONFIGLOC *logging-option-file*
specifies the path and name of a file containing any options for the SAS logging facility. SASGSUB uses the App.Grid logger name with these keys:

App.Grid.JobID	specifies the job ID as returned by the grid provider.
App.Grid.JobName	specifies the job name.
App.Grid.JobStatus	specifies the job status. Possible values are Submitted, Running, or Finished.
App.Grid.JobDir	specifies the job directory name.
App.Grid.JobDirPath	specifies the full path of job directory.
App.Grid.JobSubmitTime	specifies the time at which the job was submitted.
App.Grid.JobStartTime	specifies the time at which the job started running.
App.Grid.JobEndTime	specifies the time at which the job completed.
App.Grid.JobHost	specifies the host that ran the job.

-GRIDLIBPATH *path*
the path to the shared libraries used by the utility. This value is specified in the configuration file and should not be altered.

Note: The *path* cannot contain spaces, tabs, or any of these characters:

> < & | ^ , ; =

-GRIDSETPERMS
specifies that user directories that are created by the SAS Grid Manager Client Utility are created with Read, Write, and Execute permissions only for the user for whom the directory is created. Other users cannot access the directory. If this parameter is not specified, the operating system default permissions are used.

Alias SETPERMS

-VERBOSE
specifies that extra debugging information is printed. If this argument is not specified, only warning and error messages are printed.

Alias TRACE

-GRIDUSER *grid-logon-password* | _PROMPT_
specifies the user name to be used to log on to the grid, if required by the grid provider. This option is not required if the grid uses Platform Suite for SAS. Specify _PROMPT_ for *grid-logon-username* to specify that the SAS Grid Manager Client Utility prompts the user for a user ID.

-GRIDPASS *grid-logon-password* | _PROMPT_
specifies the password to log on to the grid, if required by the grid provider. This option is not required if the grid uses Platform Suite for SAS. Specify _PROMPT_ for *grid-logon-password* to specify that the SAS Grid Manager Client Utility prompts the user for a password.

Alias GRIDPASSWORD, GRIDPWD

-GRIDPLUGINPATH *grid-jar-file-path1 … grid-jar-file-pathN*
specifies a list of paths to search for additional grid provider JAR files. Paths are separated by semicolons. This option is not required if the grid uses Platform Suite for SAS.

Note: The *path* cannot contain spaces, tabs, or any of these characters:

> < & | ^ , ; =

Alias PLUGINPATH

-JREOPTIONS *java-runtime-options*
specifies any Java run-time options that are passed to the Java Virtual Machine. This argument is required if you are using a grid provider other than Platform Suite for SAS. This option is stored in the configuration file that is automatically created by the SAS Deployment Wizard.

-SSLCALISTLOC *filename*
specifies the location of a single file that contains the public certificates for all of the trusted certificate authorities (CA) in the trust chain. Use this option if you are using either:

- a SAS Workload Orchestrator grid with TLS/SSL enabled
- a Hadoop cluster with TLS/SSL enabled

This option is used only in UNIX environments.

-SSLCERTISS *issuer*
specifies the name of the issuer of the digital certificate that should be used by TLS. Use this option if you are using either:

- a SAS Workload Orchestrator grid with TLS/SSL enabled
- a Hadoop cluster with TLS/SSL enabled

This option is used only in UNIX environments.

-SSLCERTLOC *filename*
specifies the location of a file that contains a digital certificate for the machine's public key. Use this option if you are using either:

- a SAS Workload Orchestrator grid with TLS/SSL enabled

- a Hadoop cluster with TLS/SSL enabled

This option is used only in Windows environments.

-SSLCERTSERIAL *serial_number*
specifies the serial number of the digital certificate that should be used by TLS. Use this option if you are using either:

- a SAS Workload Orchestrator grid with TLS/SSL enabled
- a Hadoop cluster with TLS/SSL enabled

This option is used only in Windows environments.

-SSLCERTSUBJ *subject*
specifies the subject name of the digital certificate that should be used by TLS. Use this option if you are using either:

- a SAS Workload Orchestrator grid with TLS/SSL enabled
- a Hadoop cluster with TLS/SSL enabled

This option is used only in Windows environments.

-SSLCLIENTAUTH
specifies that the server should perform client authentication. Use this option if you are using either:

- a SAS Workload Orchestrator grid with TLS/SSL enabled
- a Hadoop cluster with TLS/SSL enabled

-SSLCRLCHECK
specifies that certificate revocation lists (CRLs) are checked when digital certificates are validated. Use this option if you are using either:

- a SAS Workload Orchestrator grid with TLS/SSL enabled
- a Hadoop cluster with TLS/SSL enabled

-SSLCRLLOC *filename*
specifies the location of a file that contains a CRL. Use this option if you are using either:

- a SAS Workload Orchestrator grid with TLS/SSL enabled
- a Hadoop cluster with TLS/SSL enabled

This option is used only in UNIX environments.

-SSLPKCS12LOC *filename*
specifies the location of the PKCS #12 DER encoding package file that contains the certificate and the private key. Use this option if you are using either:

- a SAS Workload Orchestrator grid with TLS/SSL enabled
- a Hadoop cluster with TLS/SSL enabled

This option is used only in UNIX environments.

-SSLPKCS12PASS *password*
specifies the password that TLS requires in order to decrypt the PKCS #12 DER encoding package file. Use this option if you are using either:

- a SAS Workload Orchestrator grid with TLS/SSL enabled
- a Hadoop cluster with TLS/SSL enabled

This option is used only in UNIX environments.

-SSLPVTKEYLOC *filename*
specifies the location of the file that contains the private key that corresponds to the digital certificate that was specified by using the SSLCERTLOC= option. Use this option if you are using either:

- a SAS Workload Orchestrator grid with TLS/SSL enabled
- a Hadoop cluster with TLS/SSL enabled

This option is used only in UNIX environments.

-SSLPVTKEYPASS *password*
specifies the password that TLS requires in order to decrypt the private key. Use this option if you are using either:

- a SAS Workload Orchestrator grid with TLS/SSL enabled
- a Hadoop cluster with TLS/SSL enabled

This option is used only in UNIX environments.

SASGSUB Command: Batch Options

These options are used when running SASGSUB in batch mode. Argument values that contain spaces must be contained in quotation marks.

Syntax

-GRIDWORK *work-directory*
<-GRIDWORKREM *shared-file-system-path*>
<-GRIDFILESIN *grid-file-list* >
<-GRIDSTAGECMD *command*>
<-GRIDSTAGEHOST *hostname*>
< -GRIDWAIT >
<-GRIDWAITTIMEOUT>*timeout-value*
<-GRIDWAITRESULTS>
<-GRIDWAITLOGLIST>
<-GRIDWAITNORESULTS>
<-GRIDWAITRESULTSNOLOG>
<-GRIDWATCHOUTPUT>

Required Argument

-GRIDWORK *work-directory*
specifies the path for the shared directory that the job uses to store the program, output, and job information. On Windows, the path can contain spaces, but must be in quotation marks. On UNIX, the path cannot contain spaces. This option is stored in the configuration file that is automatically created by the SAS Deployment Wizard.

Optional Arguments

-GRIDWORKREM *shared-file-system-path*

when using a shared file system, specifies the pathname of the GRIDWORK directory in the shared file system relative to a grid node. Use this argument when the machine used to submit the job is on a different platform than the grid. The path cannot contain spaces.

When you are using file staging, this argument specifies the location of the GRIDWORK directory as passed to the STAGECMD parameter. For example, for Windows clients, GRIDWORK could be set to `\\myServer\myShare\SharedPath` with the grid using LSRCP to stage files with GRIDSTAGEFILEHOST set to `myServer.mydomain.com`. If you set Windows to map `C:\myShare` on myServer to the Windows share `\\myServer\myShare`, you would set GRIDWORKREM to `C:\myShare\SharedPath`.

-GRIDFILESIN *grid-file-list*

specifies a comma-separated list of files that need to be moved to the grid work directory before the job starts running.

Alias FILESIN

-GRIDSTAGECMD *command*

specifies the remote copy command used to stage files to the grid. Valid values are LSRCP, RCP, SCP, or SMBCLIENT. This option is used when the grid client machines and the grid machines do not share a common directory structure.

Alias STAGECMD

-GRIDSTAGEHOST *hostname*

specifies the name of the host that stores files that are staged into the grid. This option is used when a machine submits files to the grid and then shuts down.

Alias STAGEHOST

-GRIDWAIT

specifies that the SAS Grid Manager Client Utility waits until the job has completed running, either successfully or with an error. If the job does not complete, it must be ended manually. The results of the job are not returned, and the job directory is retained on the GRIDWORK location.

Alias WAIT

-GRIDWAITTIMEOUT *timeout-value*

specifies that the SAS Grid Manager Client Utility waits until either the job has completed running (either successfully or with an error) or *timeout-value* (in seconds) is reached. If you use this option with -GRIDWAIT or -GRIDWAITRESULTS, it enables you to specify how long to wait for results.

Alias WAITTIMEOUT, TIMEOUT

-GRIDWAITRESULTS

specifies that the SAS Grid Manager Client Utility waits until the job has completed running (either successfully or with an error). It then returns the results of the processing, including the job directory and the job log and output listing. If the job does not complete, it must be ended manually.

Alias WAITRESULTS

-GRIDWAITLOGLIST
specifies that the SAS Grid Manager Client Utility waits until the job has completed running (either successfully or with an error). It then returns the results of the processing. After obtaining the job log and output listing, the job directory is deleted. If the job does not complete, it must be ended manually.

Alias WAITLOGLIST

-GRIDWAITNORESULTS
specifies that the SAS Grid Manager Client Utility waits until the job has completed running (either successfully or with an error). It then returns the results of the processing and deletes the job log, the output listing, and the job directory. If the job does not complete, it must be ended manually.

Alias WAITNORESULTS

-GRIDWAITRESULTSNOLOG
specifies that, after a job has completed running and the results of the processing are returned, the job log, output listing, and job directory are deleted, and the log and list files are not copied to the current directory.

Alias WAITRESULTSNOLOG

-GRIDWATCHOUTPUT
specifies that the output of what was submitted by the SASGSUB command is displayed on the user's machine. If this argument is used with -GRIDSUBMITPGM, the SAS log and output are displayed. If the argument is used with -GRIDRUNCMD, the command's standard output and standard error are displayed. While the output is being displayed, entering the command prompt does not affect the processing on the grid.

Alias WATCHOUTPUT

SASGSUB Command: Grid Job Arguments

These arguments are used when submitting a SAS job or command to the grid. Argument values that contain spaces must be contained in quotation marks.

Syntax

<-GRIDJOBNAME *grid-program-name*>
<-GRIDJOBOPTS *grid-provider-options* >
<-GRIDOPTSET> *options-set-name*
<-GRIDWORKLOAD *grid-resource-names*>

Optional Arguments

-GRIDJOBNAME *grid-program-name*
specifies the name of the grid job as it appears on the grid. If this argument is not specified, the SAS program name is used.

Alias JOBNAME

-GRIDJOBOPTS *grid-provider-options*

specifies any options that are passed to the grid provider when the job is submitted. You can separate multiple options with spaces or with semicolons. If you use spaces, you must enclose the option string in quotation marks. For a grid that uses SAS Grid Manager, see "Supported Job Options – SAS Workload Orchestrator" on page 305.For a grid that uses Platform Suite for SAS, see Appendix 2, "Supported Job Options - SAS Grid Manager for Platform," on page 307. For a grid that uses SAS Grid Manager for Hadoop, see Appendix 3, "Supported Job Options - SAS Grid Manager for Hadoop," on page 311.

Alias JOBOPTS

-GRIDOPTSET *options-set-name*

specifies the name of a grid options set to be used in job processing. This option is needed only if there is more than one grid options set associated with the current SAS Grid Manager Client Utility user. If there is only one grid options set for the user, it is used by default. A grid options set is a collection of grid options, SAS options, and required resources that are associated with a particular SAS client application and SAS user or group.

Alias OPTSET

-GRIDWORKLOAD *grid-resource-name*

Alias WORKLOAD

PART 7

Appendix

Appendix 1

Supported Job Options - SAS Grid Manager

Supported Job Options – SAS Workload Orchestrator

The following table lists the job options that are supported by SAS Workload Orchestrator. You can specify these options in these locations:

- the JOBOPTS= option of the GRDSVC_ENABLE function
- the **Additional Options** field in the metadata definition for the SAS Logical Grid Server
- the **Grid Options** field in the grid server definition and in grid options set definitions

Options that are specified in metadata override those that are specified in a GRDSVC_ENABLE statement.

Table A3.1 *SAS Workload Orchestrator Job Option Name/Value Pairs*

Job Option Name/Value Pairs	Explanation
authBasic=1 | 0	if set to 1, specifies that the logical grid server uses Basic authentication when communicating with SAS Workload Orchestrator.
authKerberos=1 | 0	if set to 1, specifies that the logical grid server uses Negotiate (Kerberos) authentication when communicating with SAS Workload Orchestrator.

Job Option Name/Value Pairs	Explanation
coresRequired=*value*	specifies the number of cores that are required on a host in order to run the job.
masterList=*host1,host2,...hostn*	specifies the master host candidates in the order in which they should be contacted.
memRequired=*value*	specifies the amount of memory (in MB) that is required on a host in order to run the job.
noSecurity=1 | 0	if set to 1, disables authentication when contacting the SAS Workload Orchestrator master host.
noSSL=1 | 0	if set to 1, disables TLS/SSL.
queue=*queue_name*	specifies the name of the queue to put the job in. The default queue name is `default`.
startTime=*delayed_start_time*	specifies that the job does not run until the time that is specified by the *delayed_start_time* parameter. Here are the valid forms of *delayed_start_time*: ■ *year:month:day:hour:min* ■ *month:day:hour:min* (runs in the current year) ■ *day:hour:min* (runs in the current year and month) ■ *hour:min* (runs in the current year, month, and day) Here are the valid values for the parameters: *min* 0–59 *hour* 0–23 *day* 1–31 *month* 1–12 *year* four-digit form of a year
testHost=*host*	specifies the name of the host to which a test job is sent.

Appendix 2

Supported Job Options - SAS Grid Manager for Platform

Supported Job Options – SAS Grid Manager for Platform

The following table lists the job options that are supported by SAS Grid Manager for Platform. You can specify these options in these locations:

- the JOBOPTS= option of the GRDSVC_ENABLE function
- the **Additional Options** field in the metadata definition for the SAS Logical Grid Server
- the **Grid Options** field in the grid server definition and in grid options set definitions

Options specified in metadata override those specified in a GRDSVC_ENABLE statement.

Table A4.1 *SAS Grid Manager for Platform Job Option Name/Value Pairs*

Job Option Name/Value Pairs	Explanation
app=*grid_application_name*	specifies the LSF application name to use. This option is valid only for LSF version 7 and later.
exclusive=0|1	specifies whether the job runs as the only job on the grid node. 0 means that the job does not run

Job Option Name/Value Pairs	Explanation
	exclusively; 1 means that the job runs exclusively. The default is 0.
ignoreFull=*value*	if value is anything other than 0, specifies that during object spawner load balancing, SAS Grid Manager for Platform ignores the closed status of a host if the host is closed because it is full of jobs. If the value is 0, a host that is closed because it is full is not used for the next server.
jobgroup=*job-group*	specifies the name of the job group to associate with the job. This option is equivalent to the LSF command `bsub -g`.
jobname=*job_name*	specifies a job name to be used for all jobs. The name specified in this option overrides any job name specified by an application.This option is equivalent to the LSF command `bsub -J`.
jobSlots=*number_of_slots*	specifies the number of available job slots for the grid. Use this option in configurations where the grid is controlled by Enterprise Grid Orchestrator (EGO), a component for Platform Suite for SAS. EGO always shows grid hosts as being closed. However, specifying a high value for this option shows the grid as being open and enables jobs to be submitted to the grid.
maxpendingtime=*time*	specifies the number of seconds a job can wait in the pending state before it is terminated.
priority=*job-priority*	specifies the user-assigned job priority. This is a value between 1 and MAX_USER_PRIORITY, as defined in the lsb.params file.This option is equivalent to the LSF command `bsub -sp`.
project=*projectv*	specifies the name of the project to associate with the job. This option is equivalent to the LSF command `bsub -p`.
queue=*queue*	specifies the name of the queue to put the job in. The default queue name is `normal`. This option is equivalent to the LSF command `bsub -q`.
reqres="*requested-resources*"	specifies additional resource requirements. This option is equivalent to the LSF command `bsub -R`.
runlimit=*time-in-seconds*	specifies the maximum amount of time that a job is allowed to run. This value is used as an absolute

Job Option Name/Value Pairs	Explanation
	limit or as part of an SLA job. This option is equivalent to the LSF command `bsub -W`.
sla=*service-level-agreement*	specifies the name of the service-level agreement to associate with the job. This option is equivalent to the LSF command `bsub -sla`.
startTime=*delayed_start_time*	specifies that the job does not run until the time specified by the *delayed_start_time* parameter. These are the valid forms of *delayed_start_time*: ■ *year:month:day:hour:min* ■ *month:day:hour:min* (runs in the current year) ■ *day:hour:min* (runs in the current year and month) ■ *hour:min* (runs in the current year, month, and day) These are the valid values for the parameters: *min* 0–59 *hour* 0–23 *day* 1–31 *month* 1–12 *year* four-digit form of a year This option is equivalent to the LSF command `bsub -b`.
usergroup=*user-group*	specifies the name of the user group. This option is equivalent to the LSF command `bsub -G`.

For complete information about job options, see *IBM Spectrum LSF command reference* at https://www.ibm.com/support/knowledgecenter.

Appendix 3

Supported Job Options - SAS Grid Manager for Hadoop

Supported Job Options – SAS Grid Manager for Hadoop

The following table lists the job options that are supported by SAS Grid Manager for Hadoop. You can specify these options in these locations:

- the JOBOPTS= option of the GRDSVC_ENABLE function
- the **Additional Options** field in the metadata definition for the SAS Logical Grid Server
- the **Grid Options** field in the grid server definition and in grid options set definitions

Options specified in metadata override those specified in a GRDSVC_ENABLE statement.

Table A5.1 *SAS Grid Manager for Hadoop Options*

Job Option	Explanation
appMasterMem=*memory*	specifies amount of memory in megabytes needed for the application master to run.
appType=*applicationTypeName*	specifies the application type to use when running the job. *applicationTypeName* must be defined in the SASGRID policy file. If *applicationTypeName* does not match an entry in the policy file, the

Job Option	Explanation
	application type that is specified by the defaultAppType attribute is used. See "Creating the SASGRID Policy File" on page 200.
cfgPath=*path*	specifies the absolute path to the Hadoop configuration files.
jaasCfgFile=*path*	specifies the absolute path to the JAAS configuration files.
jarFile=*path*	specifies the path to the `sas.grid.provider.yarn.jar` file.
jarPath=*path*	specifies the path to the Hadoop JAR files.
javaOpts=*options*	specifies the Java options to add to the Java command for Hadoop clusters.
krb5ccFile=*path*	specifies the absolute path to the Kerberos 5 credential cache for the user.
krb5CfgFile=*path*	specifies the absolute path to the Kerberos 5 configuration file.
noHDFS	specifies that HDFS is not used.
noMapRLogin	specifies that `maprlogin kerberos` is not automatically issued when submitting jobs to MapR clusters.
noSecurity	specifies that security is not enabled.
policyFile=*path*	specifies the absolute path to the SAS Grid policy XML file.
testHost=*hostname*	specifies a host to use for grid validation. The job is sent to the specified host using minimum resource requirements.
useJava	specifies that Java should be used, even if the Hadoop cluster supports the YARN REST API.
useLog4j	specifies that the file `hdfs://tmp/SASGrid/log4j.properties` file should be used for logging. You must copy the file to this location manually. **Note:** As of release 9.4M8, Log4J2.default release 2.17.1 is used.
useSSL	specifies that the job should use an SSL connection.

Job Option	Explanation
vjrPath	specifies the absolute path for the versioned JAR repository (VJR).

Appendix 4

Creating Cron Expressions

Writing cron Expressions

SAS Workload Orchestrator uses cron expressions to specify start and end times for time-based queue and host parameters.

Cron expressions use this format. Parameters are separated by a space.

```
min hour day-of-month month day-of-week
```

Here are the values and special characters for each parameter that are most likely to be used in SAS Workload Orchestrator:

min
: 0–59 (number of minutes)

hour
: 0–23 (number of hours)

day-of-month
: 1–31 (day of the month)

month
: 1–12 or JAN–DEC

day-of-week
: 0–6 or SUN–SAT

These special characters are also allowed in all values:

,
: separates multiple values in a list

-
: separates values in a range

*
: specifies any instance of the value (such as any *month* or *day-of-week*)

Appendix 5

Sample Scripts

Sample User-Defined Resource Scripts

Overview

The sampleResourceScript files provide examples of code for defining global static, host static, and host dynamic resources. A user-defined resource script file requires the resource name as an input parameter (in this example, myGlobalStatic, myHostStatic, or myHostDynamic), and it returns the value of the resource. See “Defining User-Defined Resources” on page 60 for more information.

UNIX

```
#!/bin/sh
##*************************************************************
##
## Example script to handle a user-defined resource
##
## Script takes one parameter, namely the resource name
## specified in the SAS Workload Orchestrator configuration
##
##*************************************************************

resName=$1

##*************************************************************
##
## Do something different based on which resource this is
## being asked about
##
##*************************************************************
case $resName in

    ##*******************************************************
    ## "My Global Static" resource. This is called on the
    ## master when it becomes the master.
    ##
    ## In this example, assume the resource will be the number
    ## of licenses of some product that can be used anywhere
    ## on the grid. We are indicating there are 20 of these
    ## licenses that can be consumed by all jobs on all hosts
    ##*******************************************************
    myGlobalStatic)
      echo "20.0"
      ;;

    ##*******************************************************
    ## "My Host Static" resource. This is called on the host
    ## whenever it announces itself to the master or when the
    ##
    ## In this example, assume the resource will be the number
    ## of GPUs available on this host. We are indicating there
    ## are 4 GPUs on this machine that can be consumed by all
    ## jobs routed to this host.
    ##*******************************************************
    myHostStatic)
      echo "4.0"
      ;;

    ##*******************************************************
    ## "My Host Dynamic" resource. This is called on the host
    ## periodically as defined by the hostUpdateSecs field in
    ## the configuration.
```

```
    ##
    ## In this example, assume the resource is the percent of
    ## used space in the SASWORK directory. We are going to
    ## return a value that will be able to be used for
    ## scheduling or suspend thresholds and for sort order.
    ##*********************************************************
    myHostDynamic)
      echo `df /path/to/SASWORK | awk '$3 ~ /[0-9]+/ { print $4 }' |
sed -e 's/%//'`
      ;;

    ##*********************************************************
    ## Unknown resource. Return 0
    ##*********************************************************
    *)
      echo "0.0"
      ;;
esac
```

Windows

The Windows sampleResourceScript.cmd file calls the sampleResourceScript.psl file.

sampleResourceScript.cmd

```
@echo off
::*************************************************************
::
:: Windows command file to invoke PowerShell to run PS script
:: since SAS Workload Orchestrator does not run PS1 files
:: directly and because if it did, the PowerShell security
:: would need to be set up to run the sampleJobChangeScript
::
::*************************************************************

set CMDPATH=%~dp0

powershell %CMDPATH%sampleResourceScript.ps1 %1
```

sampleResourceScript.psl

```
##*************************************************************
##
## Example script to handle a user-defined resource
##
## Script takes one parameter, namely the resource name
## specified in the SAS Workload Orchestrator configuration
```

```
##
##************************************************************

Param(
   [Parameter(Mandatory=$true, HelpMessage="Resource name as defined
in the
SAS Workload Orchestrator configuration.")]
   [string] $resName
)


##************************************************************
##
## Get-FreeSpace returns the megabytes free for a specified
## path
##
##************************************************************
function Get-FreeSpace
{
    param (
      $path
    );

    $free = gwmi Win32_Volume -Filter "DriveType=3" `
              | where { $path -Like "$($_.Name)*" } `
              | sort Name -Desc `
              | select -First 1 FreeSpace `
              | % { $_.freespace / (1024*1024) }

    return ([int]$free)
}

##************************************************************
##
## Do something different based on which resource this is
## being asked about
##
##************************************************************
switch($resName)
{
    ##*******************************************************
    ## "My Global Static" resource. This is called on the
    ## master when it becomes the master.
    ##
    ## In this example, assume the resource will be the number
    ## of licenses of some product that can be used anywhere
    ## on the grid. We are indicating there are 20 of these
    ## licenses that can be consumed by all jobs on all hosts
    ##*******************************************************
    'myGlobalStatic'
    {
        Write-Host "20.0" -NoNewLine
    };

    ##*******************************************************
    ## "My Host Static" resource. This is called on the host
```

```
    ## whenever it announces itself to the master or when the
    ##
    ## In this example, assume the resource will be the number
    ## of GPUs available on this host. We are indicating there
    ## are 4 GPUs on this machine that can be consumed by all
    ## jobs routed to this host.
    ##*******************************************************
    'myHostStatic'
    {
        Write-Host "4.0" -NoNewLine
    };

    ##*******************************************************
    ## "My Host Dynamic" resource. This is called on the host
    ## periodically as defined by the hostUpdateSecs field in
    ## the configuration.
    ##
    ## In this example, assume the resource is the percent of
    ## used space in the SASWORK directory. We are going to
    ## return a value that will be able to be used for
    ## scheduling or suspend thresholds and for sort order.
    ##*******************************************************
    'myHostDynamic'
    {
        $freeSpace = Get-FreeSpace("drive:\path\to\SASWORK")
        Write-Host $freeSpace -NoNewLine
    }

    ##*******************************************************
    ## Unknown resource. Return 0
    ##*******************************************************
    default
    {
        Write-Host "0.0" -NoNewLine
    }
}
```

Sample Service Scripts

Overview

The sampleServiceScript files provide examples of code for managing a high-availability service. The script file requires an action as an input parameter (start, stop, status, or restart), and it returns a return code (for start, stop, and restart) or status value (for status). See “Defining High Availability Services on a SAS Grid Manager Grid” on page 89 for more information.

UNIX

```
#!/bin/sh
##************************************************************
##
## Example script to handle a service
##
##************************************************************

action=$1

name="SampleService"
script=`basename "$0"`
now=`date +%Y-%m-%d@%H:%M:%S`
command="sleep 600"
thisHost=`hostname`
log_filename="/tmp/${name}.${thisHost}.log"
pid_filename="/tmp/${name}.${thisHost}.pid"

##************************************************************
##
## Define the functions to be used
##
##************************************************************

##*********************************************************
## Start the Service
##*********************************************************
start_service()
{
  if [ -f $pid_filename ]; then
     pid=`cat $pid_filename`
     kill -0 $pid > /dev/null 2>&1
     if [ $? -eq 0 ]; then
        echo "${now} ${script}: Service ${name} (pid $pid) is already
running"
        exit 0
     fi
     rm $pid_filename
  fi
  nohup $command > $log_filename 2>&1 &
  pid=$!
  echo $pid > $pid_filename
  echo "${now} ${script}: Service ${name} (pid $pid) is started"
}

##*********************************************************
## Stop the Service
##*********************************************************
stop_service()
{
  if [ -f $pid_filename ]; then
```

```
    pid=`cat $pid_filename`
    kill $pid > /dev/null 2>&1
    if [ $? -ne 0 ]; then
      echo "${now} ${script}: Service ${name} (pid $pid) could not be
stopped"
    else
      echo "${now} ${script}: Service ${name} (pid $pid) has been
stopped"
      rm $pid_filename
    fi
  else
    echo "${now} ${script}: Service ${name} is stopped"
    exit 1
  fi
}

##*************************************************************
## Get the Service's status
##
##   status = 0, everything is OK
##   status < 0, temp error, retry 5 times before restarting
##   status > 0, error, try restarting
##*************************************************************
get_service_status()
{
  if [ -f $pid_filename ]; then
    pid=`cat $pid_filename`
    kill -0 $pid > /dev/null 2>&1
    if [ $? -ne 0 ]; then
      echo "${now} ${script}: Service ${name} (pid $pid) is stopped"
      exit 1
    else
      echo "${now} ${script}: Service ${name} (pid $pid) is running"
      exit 0
    fi
  else
    echo "${now} ${script}: Service ${name} is assumed to be stopped"
    exit 1
  fi
}



##*************************************************************
##
## Perform the requested action
##
##*************************************************************
case $action in

  ##*************************************************************
  ## Start the Service
  ##*************************************************************
  start | -start)
    start_service
```

```
    ;;

  ##*************************************************************
  ## Stop the Service
  ##*************************************************************
  stop | -stop)
    stop_service
    ;;

  ##*************************************************************
  ## Get the service's status
  ##*************************************************************
  status | -status)
    get_service_status
    ;;

  ##*************************************************************
  ## Restart the service
  ##*************************************************************
  restart | -restart)
    echo "${now} ${script}: Service ${name} is being restarted"
    stop_service
    sleep 1
    start_service
    ;;

  ##*************************************************************
  ## Unknown option
  ##*************************************************************
  *)
    echo "Invalid option \"$1\""
    echo "Usage: $SCRIPT {-}{start|stop|status|restart}"
    exit 1
esac

exit 0
```

Windows

The Windows sampleServiceScript.cmd file calls the sampleServiceScript.psl file.

sampleServiceScript.cmd

```
@echo off
::*************************************************************
::
:: Windows command file to invoke PowerShell to run PS script
:: since SAS Workload Orchestrator does not run PS1 files
:: directly and because if it did, the PowerShell security
:: would need to be set up to run the sampleJobChangeScript
::
```

```
::*************************************************************

set CMDPATH=%~dp0

powershell %CMDPATH%sampleServiceScript.ps1 %1
```

sampleServiceScript.psl

```
##*************************************************************
##
## Example script to handle a service
##
##*************************************************************

Param(
   [Parameter(Mandatory=$true, HelpMessage="Action to take on service.
Valid values are start, stop, status, or restart.")]
   [string] $action
)

##*************************************************************
##
## Some definitions...
##
##*************************************************************
$name="SampleService"
$script=$MyInvocation.MyCommand.Name
$now=Get-Date -Format u
$exe="sleep";
$args={"60"};
$thisHost=[System.Net.Dns]::GetHostName();
$tmp=$Env:TEMP + "\";
$namebase=$tmp + $name + "." + $thisHost;
$logFilename=$namebase+".log";
$errFilename=$namebase+".err";
$pidFilename=$namebase+".pid";


##*************************************************************
##
## Service function definitions
##
##*************************************************************
#
##*********************************************************
## Start Service function
##*********************************************************
function Start-Service
{
   param([string]   $now,
         [string]   $name,
         [string]   $cmd,
         [string[]] $cmdArgs,
         [string]   $logFileName,
```

```
            [string]   $errFilename)

   $app = Start-Process -FilePath $cmd `
                        -ArgumentList $cmdArgs `
                        -RedirectStandardOutput $logFilename `
                        -RedirectStandardError $errFilename `
                        -PassThru `
                        -ErrorVariable myErr `
                        -ErrorAction SilentlyContinue;

   if ($app)
   {
      $app.Id | Out-File $pidFilename;

      Write-Host "$now Started service '$name' with process ID $app.Id"

      $errVal = 0;
   }
   else
   {
      Write-Host "$now Service '$name' failed to start"

      $myErr;

      $errVal = 1;
   }

   return($errVal);
}

##*************************************************************
## Stop Service function
##*************************************************************
function Stop-Service
{
   param([string] $now,
         [string] $name,
         [string] $pidFilename)

   $myPid = Get-Content $pidFilename `
                        -ErrorVariable myErr `
                        -ErrorAction SilentlyContinue;

   if (!$myErr)
   {
      Stop-Process -Id $myPid -ErrorAction SilentlyContinue;

      Write-Host "$now Service '$name' with ID $myPid has been stopped"

      Remove-Item $pidFilename;

      $errVal = 0;
   }
   else
   {
      Write-Host "$now Service '$name' is not running"
```

```
      $errVal = 1;
   }

   return ($errVal);
}


##*************************************************************
##
## Main action handler
##
##*************************************************************
switch ($action)
{
  ##*********************************************************
  ## Start service
  ##*********************************************************
  'start'
  {
     $exitCode = Start-Service -now $now `
                               -name $name `
                               -cmd $exe `
                               -cmdArgs $args `
                               -logFilename $logFilename `
                               -errFilename $errFilename;
  };

  ##*********************************************************
  ## Stop service
  ##*********************************************************
  'stop'
  {
     $exitCode = Stop-Service -now $now `
                              -name $name `
                              -pidFilename $pidFilename;
  };

  ##*********************************************************
  ## Get service status
  ##   status = 0, everything is OK
  ##   status < 0, temp error, retry 5 times before restarting
  ##   status > 0, error, try restarting
  ##*********************************************************
  'status'
  {
     $myPid = Get-Content $pidFilename -ErrorVariable myErr -
ErrorAction
SilentlyContinue;

     if (!$myErr)
     {
         $app = Get-Process -Id $myPid -ErrorVariable myErr -
ErrorAction
SilentlyContinue;
```

```
            if (!$app)
            {
                Write-Host "$now Service '$name' with ID $myPid is not
running"

                $exitCode = 1
            }
            else
            {
                Write-Host "$now Service '$name' with ID $myPid is
running"

                $exitCode = 0
            }
        }
        else
        {
            Write-Output "$now Service '$name' is not running"

            $exitCode = 1
        }
    };

    ##*********************************************************
    ## Restart service
    ##*********************************************************
    'restart'
    {
       Stop-Service -name $name -pidFilename $pidFilename;

       sleep 1;

       $exitCode = Start-Service -name $name -exe $exe -args $args
-logFilename $logFilename -errFilename $errFilename;
    };

    ##*********************************************************
    ## Bad input
    ##*********************************************************
    default
    {
       Write-Host "Invalid option: '$action'";
       Write-Host "Usage: $script [-action] {start|stop|status|restart}";
       $exitCode = 2
    }
}

Write-Host "Exiting with rc=$exitCode"

exit $exitCode
```

Sample Job Change Scripts

Overview

The sampleJobChangeScript files enable a job's information to be changed when a job is submitted to the grid. The input to the script is the job's JSON in UTF-8 encoding, which is passed to STDIN. The script should return the job's modified JSON to STDOUT in UTF-8 encoding.

UNIX

```
#!/bin/sh
##************************************************************
##
## Example script to perform changes to a job request
##
## Code assumes OS is running in UTF-8 encoding
##
##************************************************************

#
# Try to set encoding to UTF-8
#
export LANG=C.UTF-8

#
# Read the JSON from STDIN (assumes OS encoding is UTF-8)
#
read -r json

#
# Change what you want to change
#
newJson=`echo $json | sed -e 's/default/priority/'`

#
# Output the JSON to STDOUT (assumes OS encoding is UTF-8)
#
echo $newJson
```

Python

```
#!/usr/bin/python
##*************************************************************
##
## Example script to perform changes to a job request
##
## Python defaults to UTF-8 but if you do not have that set
## PYTHONIOENCODING to UTF-8
##
##*************************************************************
import json
import sys
import codecs

#
# Read the job JSON from STDIN
#
job_json_text = sys.stdin.readline()

#
# Convert to JSON object
#
job_json = json.loads(job_json_text)

#
# Convert the JSON however you want
#
job_json['request']['queue']="priority"

#
# Output new JSON to STDOUT
#
print json.dumps(job_json, indent=4)
```

Windows

The Windows sampleJobChangeScript.cmd file calls the sampleJobChangeScript.psl file.

sampleJobChangeScript.cmd

```
@echo off
::*************************************************************
::
:: Windows command file to invoke PowerShell to run PS script
:: since SAS Workload Orchestrator does not run PS1 files
```

```
:: directly and because if it did, the PowerShell security
:: would need to be set up to run the sampleJobChangeScript
::
::************************************************************
set CMDPATH=%~dp0

:: Read the JSON from STDIN of this command script
set /p inputLine=""

:: Pipe the JSON to PowerShell's STDIN which will pipe it to the PS
script's STDIN
echo %inputLine% | powershell -command "$Input | %CMDPATH
%sampleJobChangeScript.ps1"
```

sampleJobChangeScript.psl

```
##************************************************************
##
## Example script to perform changes to a job request
##
##************************************************************

#
# Set input and output encoding to UTF-8
#
$OutputEncoding = [console]::InputEncoding = [console]::OutputEncoding
=
[System.Text.UTF8Encoding]::new()

#
# Read in the UTF-8 JSON from STDIN
#
$jobRequest = $Input | ConvertFrom-Json

#
# Modify the fields you want to modify
#
if ($jobRequest.request.queue -eq "default")
{
   $jobRequest.request.queue = "priority"
}

#
# Convert the object back to JSON
#
$jsonOutput = $jobRequest | ConvertTo-Json -Depth 8

#
# Output the object to STDOUT in UTF-8
#
Write-Host $jsonOutput -NoNewLine
```