Installation Instructions for Platform Suite for SAS® Version 9.1 for UNIX

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *Installation Instructions for Platform Suite for SAS® Version 9.1 for UNIX*. Cary, NC: SAS Institute Inc.

Installation Instructions for Platform Suite for SAS® Version 9.1 for UNIX

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513-2414.

September 2016

Table of Contents

Chapter 1 - Introduction

This document describes how to install the Platform Suite for SAS, version 9.1, on UNIX/Linux hosts for use with SAS products and solutions. The Platform Suite for SAS can be an individual addition to several SAS products and solutions to provide enterprise-level scheduling capabilities on a single server environment. The Platform Suite for SAS is also included as part of the SAS Grid Manager product to enable:

- distributed enterprise scheduling
- workload balancing
- parallelized workload balancing

The Platform Suite for SAS includes the following components:

- **Process Manager** – the interface used by the SAS scheduling framework to control the submission of scheduled jobs to LSF (Load Sharing Facility) which manages any dependencies between the jobs. The Flow Manager and Calendar Editor clients are included with Process Manager and may be optionally installed. These clients are not required by SAS; however, they do provide additional functionality.
  - **Flow Manager** - provides a visual representation of flows that have been created for a Process Manager Server. These include flows that were created and scheduled in SAS Management Console's Schedule Manager, as well as reports that have been scheduled through SAS Web Report Studio. Platform Flow Manager provides information about each flow's status and associated dependencies. You can view or update the status of jobs within a flow, and you can run or rerun a single job regardless of whether the job failed or completed successfully.
  - **Calendar Editor** - a scheduling client for a Process Manager Server. This client enables you to create new calendar entries for time dependencies for jobs that are scheduled to run on the server. You can use it to create custom versions of the calendars that are used to create time dependencies for jobs.
- **LSF** – 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.
- **Grid Management Services** – provides the run-time information about jobs, hosts and queues for display in the SAS Grid Manager Plug-in for SAS Management Console.

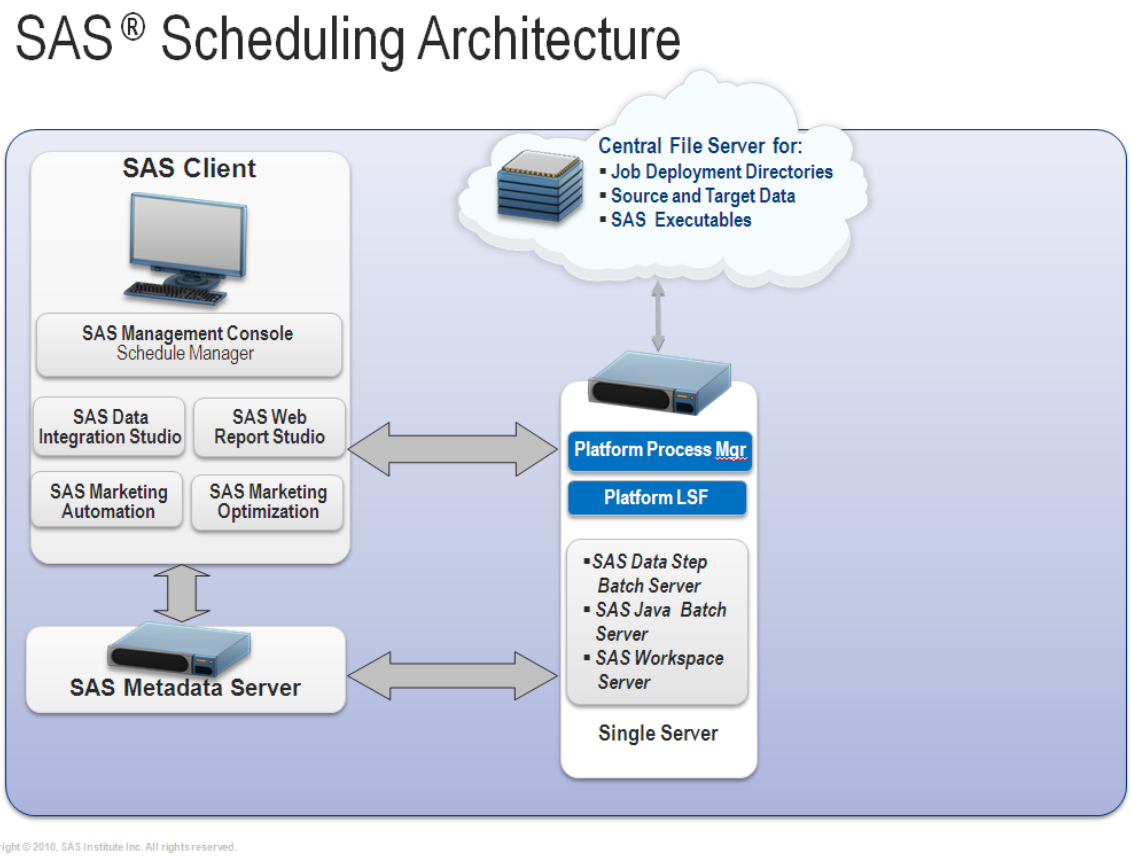
Platform RTM for SAS

Platform RTM (Report, Track & Monitor) for SAS is a web-based tool that provides IT administrators a way to graphically view the status of devices and services within their SAS grid environment as well as manage the policies and configuration of their grid. It is a visual tool to quickly track and diagnose issues before they affect service levels. Platform RTM for SAS includes drill-down capabilities to view details of hosts, jobs, queues, and user activities while instant alerts on job performance and grid efficiency allow administrators to optimize usage and workloads. It includes customizable graphs to visually analyze resource usage, workload trends, and job behavior. It also includes GUI interfaces to allow administrators to update the policies and rules in the grid configuration. Administrators can also set up high availability for any of the

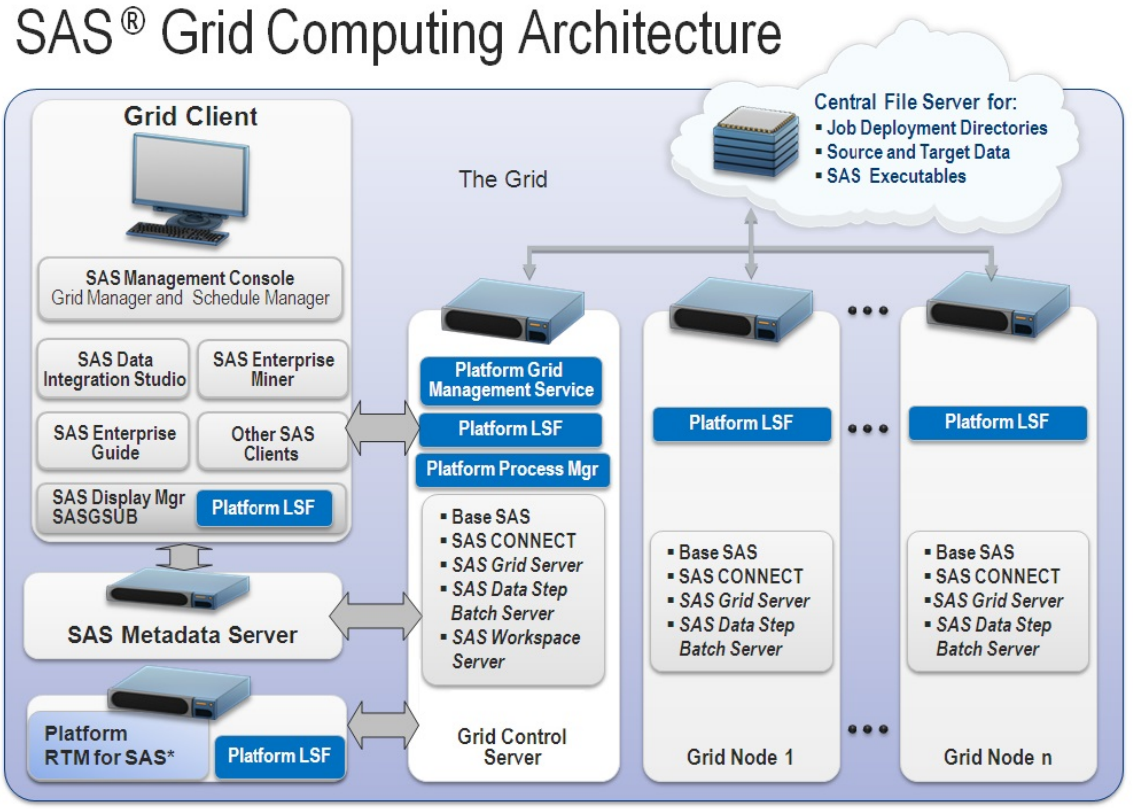
services that are critical to the operation of the grid as well as the applications executing in the grid. Platform RTM for SAS helps system administrators improve decision-making, reduce costs and increase service levels for SAS grid deployments.

You can download Platform RTM for SAS from the SAS Demos and Downloads site at `http://support.sas.com/downloads/package.htm?pid=669`. System requirements and installation instructions for Platform RTM for SAS are provided on the download page and are therefore not covered in this document.

Architecture



Job scheduling on a single machine requires that you install Platform Process Manager 9.13. During the Process Manager install, you are to install Platform LSF 9.13. See "Chapter 2 - Installing Process Manager and LSF" for the instructions on installing on a single server.



Several types of machines make up a SAS grid environment. These machines have been defined to clarify the software components that must be installed on each one as well as the SAS metadata that must be configured. The SAS Metadata Server is shown on a separate machine in this sample architecture. It is common to dedicate a machine to running the SAS Metadata Server, but you may choose to run the metadata server on the grid control server. The three machine types specific to a grid installation are defined as follows:

- **grid client** - a grid client submits work to the grid but is not part of the grid resources available to execute work. Examples of a grid client include:
  - a SAS Data Integration Studio client (Platform LSF not installed on this client machine)
  - a SAS Enterprise Miner client (Platform LSF not installed on this client machine)
  - a SAS Management Console client using the Schedule Manager plug-in or any other applications scheduling SAS workflows (Platform LSF not installed on this client machine)
  - a SAS Foundation install (minimum Base SAS, SAS/CONNECT, and Platform LSF) used to run a program that submits work—both whole programs or programs separated into parallel segments which are programs separated into parallel segments to the grid. Installation of the Platform LSF component is required in this case in order for SAS/CONNECT to submit the work to the grid.
  - Platform RTM for SAS (requires Platform LSF)
- **grid control server** - any machine in the grid can be designated as the grid control server. More software is installed on the grid control server and more SAS metadata configuration

takes place on this machine. You are to start the installation of the Platform Suite for SAS on this machine. In a SAS Data Integration Studio and SAS Enterprise Miner scenario the grid control server runs a workspace server that executes programs that utilize SAS/CONNECT to distribute work to the grid nodes. The grid control server can be configured as a grid resource capable of receiving work to execute or not, depending on the needs of your environment.

- **grid node** - a grid node is a grid computing resource capable of receiving the work that is being distributed. Each grid node must be running a minimum of Base SAS, SAS/CONNECT and Platform LSF.

Installation of Platform Suite for SAS is performed first on the grid control server and is followed by installation on all of the grid node machines. Installation of Platform LSF on the grid control server can be installed as a part of the Process Manager installation or it can be installed by itself. This document only shows LSF being installed as part of the Process Manager installation.

Machines that do processing for the grid as well as machines that submit jobs to run on the grid must have Platform LSF installed. Grid clients such as SAS Data Integration Studio or SAS Enterprise Miner do not submit jobs directly but rather work with a SAS workspace server or a stored process server that does the job submission. Since those grid clients do not submit jobs, they do not need Platform LSF installed, but the machine where the workspace server or stored process server would need it installed. If you are writing your own grid-enabled SAS program in SAS Foundation and want to run the program, that grid client workstation must have Platform LSF installed since it processes the actual submission of jobs to the grid.

This document is designed to assist you with installing Platform Suite for SAS to create a computer cluster and enable the cluster to work with the SAS Business Intelligence Platform. Please refer to the *Grid Computing in SAS document*, located at
`http://support.sas.com/documentation/onlinedoc/gridmgr/index.html.`

SAS Grid Manager Control Server requires Platform Process Manager 9.13 and Platform Grid Management Service 8.01. Platform LSF 9.13 is to be installed during the Platform Process Manager 9.13 installation. See “Chapter 2 - Installing Process Manager and LSF” for the instructions on installing Process Manager and LSF. See “Chapter 3 - Installing Grid Management Service (GMS)” for instructions on installing Grid Management Services. SAS Grid Manager Node and SAS Grid Manager Client require only Platform LSF 9.13. See “Chapter 4 - Installing LSF on Grid Nodes, SAS Foundation Grid Clients” for the instructions on installing LSF.

Installation Directories

Caution: *Do not install Process Manager and LSF to the same directory.*

Installing Platform Suite for SAS produces the following directory structure:

- `JS_TOP` is the local directory in which the Process Manager Server and Client files are installed, (for example, `/usr/share/pm`).
- `LSF_TOP` is the shared directory in which LSF files are installed (for example, `usr/share/lsf`). Generally, `LSF_TOP` is mounted from a file server and all files associated with LSF (state files, binaries for the different architectures, configuration files) are stored in this file share. `LSF_TOP` must be a shared directory between all the machines in the grid. For increased high availability this file server could be a machine that is not part of the grid.

- The GMS (Grid Management Service) files are installed in the LSF directory structure under their own directory, `gms` (for example, `/usr/share/lsf/gms`).
- Machine-dependent files are installed under `LSF_TOP/9.1/platform_name`. These directories and the files underneath represent the machine-dependent files. Machine-dependent files are specific to a particular host type and are the LSF command binaries, server daemons, libraries, and utilities.
- Machine-independent files are independent of the host type and are shared by all host types (main pages, configuration files, include files, examples, etc.)

Pre-Installation Requirements

1. Read the Platform Web Services deployment documentation.
2. Ensure that the SAS install user account exists. If not, create it. The user account **sas** will be used throughout this document as an example where a specific user account name is necessary for clarity. It is recommended that the same operating system account be used to deploy both SAS and Platform LSF. Separate accounts *can* be used but doing so means that a patch will be required in order to run Platform Web Services. See the Platform Web Services deployment documentation for details.
3. Contact your system administrator to create a network share that all computers on your cluster can access. This can be an NFS mount, a directory on a SAN, an SMBFS/CIFS mount, or any other method of creating a directory that is shared among all the machines in the grid. All machines in the grid must be able to access this share at boot time, so have your system administrator set that up based on the type of share. All machines in the grid must also have root access, since any machine may become the grid master. All grid users must be able to read the share at runtime. This is referred to as `LSF_TOP`, the LSF top-level installation directory. This installation concludes that `LSF_TOP` is mounted on each machine as `/usr/share/lsf`. Make sure `root` has read/write access to this subdirectory.
4. Create a directory on the grid control server that contains the Process Manager files. This is referred to as `JS_TOP`, the Process Manager top-level installation directory. This installation concludes that `JS_TOP` is `/usr/share/pm`.
5. Make a list of the names of all the computers that participate in the cluster.
6. Choose a name for the cluster.
7. Locate the SAS9*_*.txt and `LSF9*_*.txt` file located in the `sid_files` directory in your SAS Software Depot. The LSF9*_*.txt file is to license all components of Platform Suite for SAS in a scheduling capabilities on a single server environment. The SAS9*_*.txt is to license all the components of Platform Suite for SAS as part of the SAS Grid Manager.
8. Determine the types of all computers in the grid using **uname –a** so that the correct tar files can be copied for the install. This allows you to determine the subdirectory in your SAS Software Depot that contains the software for all the UNIX operating system types and CPU architectures in the grid. The files are in the `third_party` directory of the SAS Software Depot. The install requires a `pm9*.tar` of the same operating system and CPU architecture as the single machine or grid control server. The pm9* files are in `Platform_Process_Manager/9_13` directory under the sub-directory named for the host you are installing on. The install requires a `gms8.0.1_install.tar.Z` and the `gms8.0.1_*.tar.Z` for the OS/CPU type as the grid control server. The gms8.0.1_* files are in `Platform_Grid_Management_Service/8_01D`

directory under the sub-directory named for the host you are installing on. The install requires the `lsf9.1.3_lsfinstall.tar.Z` and `lsf9.1.3*.tar.Z` files for all other grid nodes OS/CPU types. The lsf9.1.3* files are in `Platform_LSF/9_13` directory under the sub-directory named for the host you are installing on. As an example, if the grid control server is a Solaris 10 SPARC system, but the grid contains Linux 2.6 machines with x86, and Linux 2.6 x86-64 machines, the following tar files are needed:

- `Platform_Process_Manager/9_13/64-bit_Enabled_Solaris/pm9.1.3.0_sas_sparc-sol10-64.tar` (which contains `lsf9.1.3_sparc-sol10-64.tar.Z`)
- `Platform_Grid_Management_Service/8_01D/64-bit_Enabled_Solaris/gms8.0.1_install.tar.Z`
- `Platform_Grid_Management_Service/8_01D/64-bit_Enabled_Solaris/gms8.0.1_sparc-sol10-64.tar.Z`
- `Platform_LSF/9_13/Linux/lsf9.1.3_lsfinstall.tar.Z`
- `Platform_LSF/9_13/Linux/lsf9.1.3_linux2.6-glibc2.3-x86.tar.Z`
- `Platform_LSF/9_13/Linux_for_x64/lsf9.1.3_linux2.6-glibc2.3-x86_64.tar.Z`

9. If your operating system performs user authentication against an LDAP/PAM server, you are required to make some configuration changes to allow GMS and PM to use the PAM interface. The instructions for making those configuration changes are located in the appendix.
10. LSF uses RSH/RLOGIN by default to execute certain commands on machines in the grid. If you would prefer to use SSH, please refer to the "Encrypt transmission of LSF commands for remote execution and login" section of the Platform LSF Security document for a list of required changes.
11. On Linux, Platform PM requires the libstdc++.so.5 library. If this is not found on your system, install libstdc++ or the compat-libstdc++ RPM containing the libstdc++.so.5 libstdc++ library that is appropriate for your system. Contact Redhat, www.redhat.com for assistance in obtaining this library information if necessary.

Chapter 2 - Installing Process Manager and LSF

Caution: *Do not install Process Manager and LSF to the same directory.*

1. Log on to the machine as the Primary LSF administrator.
2. Create a Process Manager install directory (such as `/local/pm_install`) to hold the installation files.
3. Copy the `pm9.1.3.0_sas_*.tar` file from the appropriate location (see step 6 from the pre-installation requirements section) to the install directory. The pm9.1.3 files contain both LSF and PM media. If this is part of a grid installation and there are machines in the grid that are a different operating system or CPU architecture than the grid control server, then copy the appropriate **`lsf9.1.3_*_tar.Z`** files for those operating systems/CPUs.
4. Change the working directory to the Process Manager install directory.
5. Extract the `pm9.1.3.0_sas_*.tar` file. For example, under Linux you can use the command `tar xvf pm9.1.3.0_sas_linux2.6-glibc2.3-x86.tar`. This is designed to create a `pm9.1.3.0_sas_pinstall` subdirectory in the Process Manager install directory.
6. Change to the `pm9.1.3.0_sas_pinstall` subdirectory.
7. Copy the license file obtained from SAS (for example, 'LSF*' in the `sid_files` directory in your SAS Software Depot) into the current directory and rename it `license.dat`.
8. Edit the `install.config` file and change the following sections:

Required Section	Description
`JS_TOP`	Directory to install Process Manager. This does not need to be a network share but recommended. For example: `JS_TOP=/usr/share/pm`
`JS_HOST`	Host that is assigned to be the Process Manager host. Specify the machine's fully qualified domain name (FQDN). For a grid installation this is in place to be the grid control server.

`JS ADMINS`	Space-delimited list of user IDs that will serve as LSF administrators. It is recommended that the SAS install account be used as the Primary LSF administrator, which is the first name in the list. Examples:
	`JS ADMINS=sas`
	`JS ADMINS="sas_admin2_admin3"`

`LSF_INSTALL`	Flag indicating whether to install LSF. This must be "true".
`LSF_TOP`	Network share containing the LSF installation mentioned in the pre-installation requirements. For example: `LSF_TOP=/usr/share/lsf`
`LSF_CLUSTER_NAME`	Name of cluster. For example: `LSF_CLUSTER_NAME=sas_cluster`
`LSF_MASTER_LIST`	List of servers that are going to participate as master candidates.

Required Section	Description
	The first server in the list is considered the default LSF master machine. The master list should be in the order of how the user prefers the cluster master succession should be listed. For example, HOSTA, HOSTB, and HOSTC means that HOSTA is the default master. When HOSTA expires, HOSTB will be the master and so on. The LSF master machine is the grid control server for a grid installation and the machine containing Process Manager.
`LSF_ADD_SERVERS`	List of servers that are going to participate in the cluster that are **not** required to be included as master candidates.
`LSF_ADD_CLIENTS`	List of hosts that only submit jobs to the grid.

9. Optional sections can be specified as shown below:

Optional Section	Description
`JS_PORT`	Process Manager port number. Use default of 1966 unless it is in use by another program.
`JS_TARDIR`	Path of directory to Process Manager distribution files. If not used, the tar files are expected to be in the current directory running **jsinstall**.
`JS_LICENSE`	Full path to Process Manager/LSF license file. If not used, the `license.dat` file is expected to be in the current directory running **jsinstall**.
`JS_MAILHOST`	The name of the mail server host if you want to receive email notices from LSF. For example: `JS_MAILHOST=[SMTP|Exchange:]hostname`
`LSF_TARDIR`	Path of directory to architecture specific tar files. If not used, the tar files are expected to be in the current directory running **jsinstall**.

The resulting file contains content similar to this example:

```
# -------------------------
JS_TOP=/usr/share/pm
# -------------------------
# REQUIRED. You must uncomment this keyword and specify a value.

# -------------------
JS_HOST=myhost
# -------------------
# REQUIRED. You must uncomment this keyword and specify a value.

# --------------
# JS_PORT=
# --------------
# OPTIONAL. The default port number is 1966.

# ---------------------
JS ADMINS=sas
# ---------------------
# REQUIRED. You must uncomment this keyword and specify a value.

# ----------------
LSF_INSTALL=true
# ----------------
# REQUIRED. You must uncomment this keyword and specify a value.

# -----------------
LSF_TOP="/usr/share/lsf"
# -----------------
# REQUIRED. You must uncomment this keyword and specify a value.

# -----------------
LSF_CLUSTER_NAME="sas_cluster"
# -----------------
# REQUIRED. You must uncomment this keyword and specify a value.

# -----------------
LSF_MASTER_LIST="myhost"
# -----------------
```

10. Change to the **root** user and execute the command `./jsinstall –f install.config`. Doing so installs LSF first, and then, Process Manager. A directory is created for each component and an `Install.log` file is created in each directory. All the events of the installation are logged here.

```
$ ./jsinstall -f install.config

Starting jsinstall...

Verifying the working directory...

Reading configuration file...

Updating JS and LSF config files.
This may take a few minutes...
```

11. Read and agree to the LSF End User License Agreement.

```
International Program License Agreement

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON
AN "ACCEPT" BUTTON, OR OTHERWISE USING THE PROGRAM,
LICENSEE AGREES TO THE TERMS OF THIS AGREEMENT. IF YOU ARE
ACCEPTING THESE TERMS ON BEHALF OF LICENSEE, YOU REPRESENT
AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND LICENSEE
TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

Press Enter to continue viewing the license agreement, or
enter "1" to accept the agreement, "2" to decline it, "3"
to print it, "4" to read non-IBM terms, or "99" to go back
to the previous screen.
1
```

12. When asked which architecture specific tar files to install, include all tar files for all OS/CPU machine types in your cluster.

```
LSF pre-installation check ...

Checking the LSF TOP directory /usr/share/lsf ...
... Done checking the LSF TOP directory /usr/share/lsf ...

You are installing IBM Platform LSF - 9.1.3 Standard Edition.

Checking LSF Administrators ...
   LSF administrator(s):       "sas"
   Primary LSF administrator:  "sas"

Checking the patch history directory  ...
Creating /user/share/lsf/patch ...
... Done checking the patch history directory /user/share/lsf/patch ...

Checking the patch backup directory ...
... Done checking the patch backup directory /user/share/lsf/patch/backup ...


Searching LSF 9.1.3 distribution tar files in /local/pm-
install/pm9.1.3.0_sas_pinstall Please wait ...

  1) linux2.6-glibc2.3-x86_64

Press 1 or Enter to install this host type:  1

You have chosen the following tar file(s):
    Lsf9.1.3_linux2.6-glibc2.3-x86_64

Checking selected tar file(s) ...
... Done checking selected tar file(s).
```

13. Wait while the install unpacks the architecture specific files, creates the LSF working directories, adds server hosts, configures the cluster, configures the license file, and creates the `lsf_getting_started.html` and `lsf_quick_admin.html` files.

```
Pre-installation check report saved as text file:
/install/pm_install/pm9.1.3.0_sas_pinstall/lsf9.1.3_lsfinstall/prechk.rpt.
```

```
... Done LSF pre-installation check.

Installing LSF binary files " lsf9.1.3_linux2.6-glibc2.3-x86_64"...
Creating /usr/share/lsf/9.1 ...

Copying lsfinstall files to /usr/share/lsf/9.1/install
Creating /usr/share/lsf/9.1/install ...
Creating /usr/share/lsf_/9.1/install/scripts ...
Creating /usr/share/lsf/9.1/install/instlib ...
Creating /usr/share/lsf/9.1/install/patchlib ...
Creating /usr/share/lsf/9.1/install/lap ...
Creating /usr/share/lsf/9.1/install/conf_tmp1 ...
... Done copying lsfinstall files to /usr/share/lsf/9.1/install

Installing linux2.6-glibc2.3-x86_64 ...

Please wait, extracting lsf9.1.3_linux2.6-glibc2.3-x86_64 may take up to a few minutes ...

... Adding package information to patch history.
... Done adding package information to patch history.
... Done extracting /install/pminstall/pm9.1.3.0_sas_pinstall/lsf9.1.3_linux2.6-glibc2.3-x86_64.tar.Z....

Creating links to LSF commands ...
... Done creating links to LSF commands ...

Modifying owner, access mode, setuid flag of LSF binary files ...
... Done modifying owner, access mode, setuid flag of LSF binary files ...

Creating the script file lsf_daemons ...
... Done creating the script file lsf_daemons ...

... linux2.6-glibc2.3-x86_64 installed successfully under /usr/share/lsf/9.1.

... Done installing LSF binary files "linux2.6-glibc2.3-x86_64".

Creating LSF configuration directories and files ...
Creating /usr/share/lsf/work ...
Creating /usr/share/lsf/log ...
Creating /usr/share/lsf/conf ...
Creating /usr/share/lsf/conf/lsbatch ...
... Done creating LSF configuration directories and files ...

Creating a new cluster "sas_cluster" ...
Adding entry for cluster sas_cluster to /usr/share/lsf/conf/lsf.shared.
Installing lsbatch directories and configurations ...
Creating /usr/share/lsf/conf/lsbatch/sas_cluster ...
Creating /usr/share/lsf/conf/lsbatch/sas_cluster/configdir ...
Creating /usr/share/lsf/work/sas_cluster ...
Creating /usr/share/lsf/work/sas_cluster/logdir ...
Creating /usr/share/lsf/work/sas_cluster/live_confdir ...
Creating /usr/share/lsf/work/sas_cluster/lsf_indir ...
Creating /usr/share/lsf/work/sas_cluster/lsf_cmddir ...
Updating PRODUCTS line in /usr/share/lsf/conf/lsf.cluster.sas_cluster ...
Setting common HPC external resources to /usr/share/lsf/conf/lsf.shared
Enabling LSB_SHORT_HOSTLIST in /usr/share/lsf/conf/lsf.conf ...
Enabling schmod_aps in /usr/share/lsf/conf/lsbatch/sas_cluster/configdir/lsb.modules ...
Adding default HPC queues to /usr/share/lsf/conf/lsbatch/sas_cluster/configdir/lsb.queues ...
Setting LSF_VPLUGIN path in /usr/share/lsf/conf/lsf.conf ...
```

```
Setting LSF_ASPLUGIN path in /usr/share/lsf/conf/lsf.conf ...
Setting LSF_BMPLUGIN path in /usr/share/lsf/conf/lsf.conf ...
Setting LSF_CPUSETLIB path in /usr/share/lsf/conf/lsf.conf ...
Enabling LSF_ENABLE_EXTSCHEDULER in /usr/share/lsf/conf/lsf.conf ...
Setting LSB_RLA_PORT in /usr/share/lsf/conf/lsf.conf ...
Setting LSB_CPUSET_BESTCPUS in /usr/share/lsf/conf/lsf.conf ...
Enabling schmod_cpuset in /usr/share/lsf/conf/lsbatch/sas_cluster/configdir/lsb.modules ...
Adding default HPC queues to /usr/share/lsf/conf/lsbatch/sas_cluster/configdir/lsb.queues ...
Setting required HPC external resources to /usr/share/lsf/conf/lsf.shared and
/usr/share/lsf/conf/lsf.cluster.sas_cluster...
Adding LSB_SUB_COMMANDNAME=Y to /usr/share/lsf/conf/lsf.conf ...
Adding LSB_SUB_COMMANDNAME=Y to /usr/share/lsf/conf/lsf.conf ...

Adding server hosts ...

Host(s) "bb04cnt08" has (have) been added to the cluster "sas_cluster".

Adding LSF_MASTER_LIST in lsf.conf file...

... LSF configuration is done.
... Creating EGO configuration directories and files ...
Creating /usr/share/lsf/conf/ego ...
Creating /usr/share/lsf/conf/ego/sas_cluster ...
Creating /usr/share/lsf/conf/ego/sas_cluster/kernel ...
Creating /usr/share/lsf/work/sas_cluster/ego ...
... Done creating EGO configuration directories and files.
Configuring EGO components...
... EGO configuration is done.

... LSF license, entitlement and inventory tag files are installed.
Creating lsf_getting_started.html ...
... Done creating lsf_getting_started.html

Creating lsf_quick_admin.html ...
... Done creating lsf_quick_admin.html

lsfinstall is done.

To complete your LSF installation and get your
cluster "sas_cluster" up and running, follow the steps in
"/tmp/sas_cluster_psfs_install.2104/pminstall/pm9.1.3.0_sas_pinstall/lsf9.1.3_lsfinstall/lsf_getting_star
ted.html".

After setting up your LSF server hosts and verifying
your cluster "sas_cluster" is running correctly,
see "/usr/share/lsf/9.1/lsf_quick_admin.html"
to learn more about your new LSF cluster.

After installation, remember to bring your cluster up to date
by applying the latest updates and bug fixes. After installation, remember to bring your cluster up to
date
by applying the latest updates and bug fixes. To do so, obtain
and install the latest package. You can always download the most
recent software updates from the Platform Web Support Portal (http://my.platform.com) or
ftp.platform.com.
```

14. After the LSF install completes, the Process Manager installation starts. You are to see the following progress window:

```
Starting JS installation...


Logging installation sequence in
/install/pminstall/pm9.1.3.0_sas_pinstall/pm9.1.3.0_install/Install.log

Searching for Process Manager tar files in
/install/pminstall/pm9.1.3.0_sas_pinstall, Please wait ...

  1) [SAS] Linux2.6-glibc2.3-x86_64 Server
  2) [SAS] Linux2.6-glibc2.3-x86_64 Client

List the numbers separated by spaces that you want to install.
(E.g. 1 3 7, or press Enter for all):
```

Type `1 2` to install the Process Manager Server and Client. Press **Enter** to continue.

15. The install extracts files, creates directories, and modifies access to files. After all operations have been performed, you see the following progress window:

```
You have chosen the following tar file(s):
    pm9.1.0.0_svr_sas_lnx26-lib23-x64
    pm9.1.0.0_clt_sas_lnx26-lib23-x64

Space required to install: 300000 kb.
Space available under /install/pm: 4491280 kb.
Do you want to continue installation? (y/n) [y] y
Info: Re-using JRE from LSF install...
International Program License Agreement
```

16. Read and agree to the LSF End User License Agreement.

```
Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON
AN "ACCEPT" BUTTON, OR OTHERWISE USING THE PROGRAM,
LICENSEE AGREES TO THE TERMS OF THIS AGREEMENT. IF YOU ARE
ACCEPTING THESE TERMS ON BEHALF OF LICENSEE, YOU REPRESENT
AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND LICENSEE
TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,


Press Enter to continue viewing the license agreement, or
enter "1" to accept the agreement, "2" to decline it, "3"
to print it, "4" to read non-IBM terms, or "99" to go back
to the previous screen.

PROOF OF ENTITLEMENT TO THE PARTY FROM WHOM IT WAS OBTAINED
FOR A REFUND OF THE AMOUNT PAID. IF THE PROGRAM WAS
DOWNLOADED, DESTROY ALL COPIES OF THE PROGRAM.

<lines deleted>
```

Press **Enter** to continue viewing the license agreement, or
enter "1" to accept the agreement, "2" to decline it, "3"
to print it, "4" to read non-IBM terms, or "99" to go back
to the previous screen.

You have completed viewing the license agreement. Enter "1"
to accept the agreement or "2" to decline it. If you choose
to decline the agreement, the installation is not
completed and you are not able to use the Program.

17. The install extracts files, creates directories, and modifies access to files. After all operations have been performed, you see the following progress window:

```
Process Manager pre-installation check ...

Checking the JS_TOP directory /usr/share/pm ...
... Done checking the JS_TOP directory /usr/share/pm ...
Checking selected tar file(s) ...
... Done checking selected tar file(s).


Checking Process Manager Administrators ...
   Process Manager administrator(s):       "sasadm sasadm2"
   Primary Process Manager administrator:  "sasadm"

Checking Process Manager Control Administrators ...

/install/pminstall/pm9.1.3.0_sas_pinstall/license.dat includes SAS license.
... Done checking the license ...

Pre-installation check report saved as text file:
/install/pminstall/pm9.1.3.0_sas_pinstall/pm9.1.0.0_install/prechk.rpt.

... Done Process Manager pre-installation check.

Installing binary files " pm9.1.3.0_svr_sas_lnx26-lib23-x64
pm9.1.3.0_clt_sas_lnx26-lib23-x64"...
Creating /usr/share/pm/9.1 ...
```

```
Copying jsinstall files to /usr/share/pm/9.1/install
Creating /usr/share/pm/9.1/install ...
Creating /usr/share/pm/9.1/install/instlib ...
... Done copying jsinstall files to /usr/share/pm/9.1/install

Installing linux2.6-glibc2.3-x86_64 Server...

Please wait, extracting pm9.1.3.0_svr_sas_lnx26-lib23-x64 may take up to 5 minutes
...

... Done extracting
/install/pminstall/pm9.1.3.0_sas_pinstall/pm9.1.3.0_svr_sas_lnx26-lib23-x64.tar.Z.

... linux2.6-glibc2.3-x86_64 Server installed successfully under /usr/share/pm/9.1.

Installing linux2.6-glibc2.3-x86_64 Client...

Please wait, extracting pm9.1.3.0_clt_sas_lnx26-lib23-x64 may take up to 5 minutes
...

... Done extracting
/install/pminstall/pm9.1.0.0_sas_pinstall/pm9.1.3.0_clt_sas_lnx26-lib23-x64.tar.Z.

... linux2.6-glibc2.3-x86_64 Client installed successfully under /usr/share/pm/9.1.

Modifying owner, access mode of binary files ...

... Done modifying owner, access mode of binary files ...

Done installing binary files ...
Creating /usr/share/pm/work/templates ...

Creating configuration directories and files ...
Creating /usr/share/pm/work/alarms ...
Creating /usr/share/pm/log ...
Creating /usr/share/pm/conf ...
... Done creating configuration directories and files ...

Adding queue unicodecmd to
/usr/share/lsf/conf/lsbatch/sas_cluster/configdir/lsb.queues

Done creating configuration directories and files ...

... Process Manager license setup is done.
Creating /usr/share/pm/work/calendar/ ...


Creating /usr/share/pm/properties/version ...
Please read /usr/share/pm/README for instructions on how
to start the Process Manager



jsinstall completed successfully. Done.
```

18. Change into the `LSF_TOP/9.1/install` subdirectory (for example, `/usr/share/lsf/9.1/install`). Run the following command to set up the proper initialization files for future reboots:

    ```
    ./hostsetup --top="/usr/share/lsf" --boot="y" --profile="y"
    --start="y"
    ```

    Note: *There are two dashes "`--`" in the options. See "Chapter 5 – LSF Quick Reference" for more information on the* `hostsetup` *command.*

```
$ # ./hostsetup --top="/usr/share/lsf" --boot="y" --profile="y" --start="y"
Logging installation sequence in /usr/share/lsf/9.1/log/Install.log

------------------------------------------------------------
    L S F    H O S T S E T U P    U T I L I T Y
------------------------------------------------------------
This script sets up local host (LSF server, client or slave) environment.

Setting up LSF server host "myhost" ...
Checking LSF installation for host "myhost" ... Done
Installing LSF RC scripts on host "myhost" ... Done
LSF service ports are defined in /usr/share/lsf/conf/lsf.conf.
Checking LSF service ports definition on host "myhost" ... Done

... Setting up LSF server host "myhost" is done
... LSF host setup is done.
$
```

Note: *Host setup does not require the profile be sourced.*

19. Type `ps –ef | grep <LSF_TOP>` and make sure all daemons are running. Note that **mbatchd** and **mschd** only run on the master machine, therefore, they may not show up.

```
$ . profile.lsf
$ lsadmin limstartup
Starting up LIM on <myhost> ...... done
$ lsadmin resstartup
Starting up RES on <myhost> ...... done
$ badmin hstartup
Starting up slave batch daemon on <myhost> ...... done
$ ps -ef | grep /usr/share/lsf
root     12910     1  0 10:47 ?        00:00:00 /usr/share/lsf/9.1/linux2.6-
glibc2.3-x86_64/etc/lim
root     12911 12910  0 10:48 ?        00:00:00 /usr/share/lsf/9.1/linux2.6-
glibc2.3-x86_64/etc/pim
root     12912 12910  0 10:48 ?        00:00:00 /usr/share/lsf/9.1/linux2.6-
glibc2.3-x86_64/etc/pem
         sas 12913 12910  0 10:48 ?        00:00:00 /usr/share/lsf/9.1/linux2.6-
glibc2.3-x86_64/etc/vemkd
         sas 12919 12913  0 10:48 ?        00:00:00 /usr/share/lsf/9.1/linux2.6-
glibc2.3-x86_64/etc/egosc
root     12926     1  0 10:48 ?        00:00:00 /usr/share/lsf/9.1/linux2.6-
glibc2.3-x86_64/etc/res
root     12930     1  0 10:48 ?        00:00:00 /usr/share/lsf/9.1/linux2.6-
glibc2.3-x86_64/etc/sbatchd
root     12934 12930  0 10:48 ?        00:00:00 /usr/share/lsf/9.1/linux2.6-
glibc2.3-x86_64/etc/mbatchd -d /usr/share/lsf/conf
         sas 12941 12934  0 10:48 ?        00:00:00 /usr/share/lsf/9.1/linux2.6-
glibc2.3-x86_64/etc/mbschd
root     12962 28334  0 10:49 pts/1    00:00:00 grep /usr/share/lsf
$
```

20. Start up the Process Manager Server. This can be done easily by rebooting the computer or doing the following as root:
    a. Set up the Process Manager environment by sourcing the `profile.js` file. This can be done by executing the following command: `. <JS_TOP>/conf/profile.js`. Please note the period '**.**' which is the command to 'source' the file.

       Note: *Since all Process Manager commands require the environment set up by sourcing the* `profile.js` *file, it is best practice to source the* `profile.js` *file in the default profile for the shell.*

    b. Start the **jfd** daemons with the command `jadmin start`.
    c. To start the **jfd** daemon at boot time, run the command `bootsetup` located in `JS_TOP/9.1/install`.
    d. Type `ps –ef | grep jfd` and make sure the daemon is running.

```
$ jadmin start
Starting up jfd ...
$ cd ../9.1/install
$ bootsetup
Logging installation sequence in /usr/share/pm/9.1/install/Install.log
Copying /etc/init.d/jstartup, /etc/init.d/rc5.d/S96jstartup and
/etc/init.d/rc4.d/K05jstartup
Installing Process Manager RC scripts on host "disuse" ... Done
... Process Manager boot setup is done.
$ ps -ef | grep jfd
          sas   16417     1  0 15:03 ?        00:00:00
/usr/share/pm/9.1/linux2.6-glibc2.3-x86_64/etc/jfd
root     16566   944  0 15:04 pts/0    00:00:00 grep jfd
$
```

Testing the Installation

Once the system has rebooted, you can follow these steps to make sure LSF on the grid control server or scheduling server is operating properly.

1. Log onto the machine as an LSF administrator or user.
2. Make sure the LSF daemons are running by executing the command `ps –ef | grep <LSF_TOP>`. This lists multiple daemons such as **lim**, **pim**, **res**, **sbatchd, mbatchd** and **mbschd**.
3. Run the command **`lsid`**. This displays the cluster name and the grid control server (LSF master machine) name. If you cannot find the lsid command, you may have to source the profile first by opening a command prompt and executing the following command:

   ```
   . <LSF_TOP>/conf/profile.lsf
   ```

   Please note the period '**.**' which is the command to 'source' the file.

   Note: *The* `hostsetup` *command automatically produces the sourced* `profile.lsf` *for each user, but if it does not, you need to source it yourself.*
4. Run the command **`lshosts`**. This displays static information about the grid control server (LSF master machine).
5. Run the command **`lsload`**. This displays dynamic information about the grid control server (LSF master machine).

6. Run the command **`bsub sleep 20`**. This submits a job to the grid control server since it is the only machine so far in the cluster.
7. Run the command **`bjobs`**. This displays the job information. As you repeat this command, you can see the job go from **PEND**, to **RUN**, to being removed from the queue. The following is sample output assuming the grid control server (LSF master machine) is **grid3.testgrid.com**.

```
$ lsid
IBM Platform LSF Standard 9.1.3, Sep 26 2014
Copyright International Business Machines Corp, 1992-2014.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by
GSA ADP Schedule Contract with IBM Corp.

My cluster name is sas_cluster
My master name is myhost
Cluster in ISV mode : SAS
$ lshosts
HOST_NAME      type    model  cpuf ncpus maxmem maxswp server RESOURCES
myhost       X86_64 Intel_EM  60.0     4 16048M  2055M    Yes (mg)
$ lsload
HOST_NAME       status  r15s   r1m  r15m   ut    pg  ls    it   tmp   swp   mem
myhost              ok   1.0   1.0   1.0  25%   0.0   1     1  365G 2043M   14G
$ bhosts
HOST_NAME          STATUS       JL/U    MAX  NJOBS    RUN  SSUSP  USUSP    RSV
myhost             ok              -      1      0      0      0      0      0
$
```

8. Set up the Process Manager environment by sourcing the `profile.js` file. This can be done by executing the following command: `. <JS_TOP>/conf/profile.js`. Please note the period '.' which is the command to 'source' the file.

   Note: *Since all Process Manager commands require the environment set up by sourcing the* `profile.js` *file, it is best practice to source the* `profile.js` *file in the default profile for the shell.*

9. Run the command **`jid`**. When prompted for username and password, provide the Primary LSF administrator credentials. This displays static information about the Process Manager Server.

```
$ jid
User name: sas
Password:
My Process Manager Server name is myhost.
IBM Platform Process Manager 9.1.3.0 Build 214388 (for SAS)
Copyright Platform Computing Inc., An IBM Company 1992-2013.
ISV mode: SAS.
$
```

10. Run the command **`flowmanager.`** This executes a client application to verify client communication to the Process Manager Server.

Note*: If you are installing Platform Suite for SAS for single machine scheduling, the task is complete at this stage and you can stop here. If you are installing Platform Suite for SAS for use with SAS Grid Manager, continue with the next chapter.*

Chapter 3 - Installing Grid Management Service (GMS)

Grid Management Service is a daemon that is used by the Grid Manager Plug-in for the SAS Management Console to display grid information. After installing this service, you can use the SAS Management Console to view grid information.

1. Log onto the grid control server as the Primary LSF administrator.
2. Create a Grid Management Service install directory to hold the install files.
3. Copy the `gms8.0.1_install.tar.Z` file from the appropriate location (see step 6 from the pre-installation requirements section) to the install directory along with the `gms8.0.1_<platform>_tar.Z` specific to the operating system and architecture of the grid control server.
4. Change the working directory to the Grid Management Service install directory.
5. Untar the `gms8.0.1_install.tar.Z` file. For example, under Linux you can use the command `gunzip –d gms8.0.1_install.tar.Z` followed by `tar xvf gms8.0.1_install.tar`. This created a `gms8.0.1_install` subdirectory in the Grid Management Service install directory.
6. Change into the `gms8.0.1_install` subdirectory.
7. Edit the `install.config` file and change the following sections:

Required Section	Description
`LSF_TOP`	Network share containing the LSF installation mentioned in the pre-installation requirements. For example: `LSF_TOP=/usr/share/lsf`.
`LSF_TARDIR`	Path of directory to architecture specific tar files. If not used, the tar files are expected to be in the parent directory of the current working directory. For example: `LSF_TARDIR=/usr/share/lsf/9.1/install`
`GABD_PORT`	Port GMS recommended to use for incoming connections. If not specified, it is 1976. ***Note:*** *This configuration variable is not commented on in the* `install.config` *file and needs to be added if the default is not wanted.*
`BOOT`	Setting `BOOT="Y"` installs a script that starts the service to start at boot time. Default is not to start at boot time.

8. Change to the root user and execute the following command: `./gmsinstall –f install.config`. This installs the Grid Management Service.

```
$ ./gmsinstall -f install.config


Logging installation sequence in /install/gms_install/gms8.0.1_install/Install.log
pre-installation check ...
uncompress is /usr/bin/uncompress

Please wait, extracting /install/gms_install/gms8.0.1_linux2.6-glibc2.3-x86_64.tar.Z
may take up to 5 minutes ...

Creating /usr/share/lsf/gms ...
Creating /usr/share/lsf/gms/log ...
... Done extracting /install/gms_install/gms8.0.1_linux2.6-glibc2.3-x86_64.tar.Z.
Install Platform Grid Management Service 8.0.1
Creating /usr/share/lsf/gms/8.0.1/misc ...

Modifying owner, access mode, setuid flag of files ...
... Done modifying owner, access mode, setuid flag of files ...

Creating script for starting daemon: /usr/share/lsf/gms/bin/gaadmin
Installation of Platform Grid Management Service 8.0.1 completed
$
```

9. Start the service by either rebooting the machine, running the script created in the `<GMS_TOP>/bin` directory, or using a service management tool available on your platform. In the example above, the following command would start the service (assuming you are logged on as 'root'):

   ```
   /usr/share/lsf/gms/bin/gaadmin start
   ```

Testing the Installation

Once SAS has been installed and configured, the SAS Grid Manager Plug-in in the SAS Management Console can be used to test that the Grid Management Service is working properly.

Chapter 4 - Installing LSF on Grid Nodes, SAS Foundation Grid Clients or UNIX

When the `install.config` file was filled out, you listed machines that are to be part of the grid. Some of these machines process jobs, some may submit jobs and some may do both. Grid nodes process jobs for the grid and can optionally submit jobs to the grid. SAS Foundation Grid Clients only submit jobs to the grid without processing grid jobs.

1. Verify that the host information is already in the LSF cluster file `LSF_CONFDIR/lsf.cluster.`*`cluster_name`*. If it is not, then edit the Host section of the cluster file to add the host. For more information about the Host section, see "Configuring lsf.cluster.cluster_name Host Section" in your *Administering Platform™ LSF™* document found in the `docs` directory of your SAS Software Depot containing Platform LSF.
2. Log onto each newly added machine as root.
3. Make sure access to the shared directory where LSF was installed is available. Also, make sure the share is available for the boot initialization process and all grid users can read the share.
4. Add the Primary LSF administrator user if this was not done before the installation process started.
5. Change into the `<LSF_TOP>/9.1/install` share directory (in our example, it is `/usr/share/lsf/9.1/install`).
6. Run the following command to set up the proper initialization files for future reboots:

   ```
   ./hostsetup --top="/usr/share/lsf" --boot="y" --profile="y"
   --start="y"
   ```

   Note: *There are two dashes "`--`" in the options. See "Chapter 5 – LSF Quick Reference" for more information on the hostsetup command.*

7. Run the following two commands on the grid control node to make the new node known:

   ```
   lsadmin reconfig
   badmin reconfig
   ```

Testing the Installation

Once the system has rebooted, you can follow these steps to make sure LSF on the cluster is operating properly.

1. Log onto the grid control server as an LSF administrator or user.
2. Run the command **lshosts**. This displays static information about the grid control server and all grid node machines.
3. Run the command **lsload**. This displays dynamic information about the grid control server and all grid node machines.
4. Run the command **bsub sleep 1000**. This submits a job to the cluster. Repeat this command once for each node in the cluster.

5. Run the command **bjobs**. This displays the job information. As you repeat this command, you see the job go from **PEND**, to **RUN**, to being removed from the queue.

The following is sample output of a homogeneous cluster where the grid control server (LSF master machine) is **myhost** running Linux and the grid nodes are **node1**.sas.com, **node2**, **node3**, and **node4**, all running Linux.

```
$ . /usr/share/lsf/conf/profile.lsf
$ lshosts
HOST_NAME      type     model  cpuf ncpus  maxmem maxswp server RESOURCES
myhost        X86_64  Intel_EM  60.0      1  3291M  2047M    Yes (mg SASApp)
node1.sas.com X86_64  Intel_EM  60.0      1   250M   511M    Yes (mg SASApp)
node2         X86_64  Intel_EM  60.0      1   250M   511M    Yes (mg SASApp)
node3         X86_64  Intel_EM  60.0      1   250M   511M    Yes (mg SASApp)
node4         X86_64  Intel_EM  60.0      1   250M   511M    Yes (mg SASApp)
$ lsload
HOST_NAME       status  r15s   r1m  r15m   ut    pg  ls    it   tmp   swp   mem
node3               ok   0.1   0.0   0.5   2%   9.3   0     8 9888G  510M  211M
node1.sas.com       ok   0.5   0.0   0.4   2%   8.2   0     7 9736G  510M  210M
node2               ok   0.6   0.7   0.8  17% 232.5   1     0 9888G  510M  168M
node4               ok   0.6   0.7   0.8  17% 230.2   1     0 9888G  510M  166M
myhost              ok   1.0   0.0   0.7  55%  16.4   1     0 8906G 2047M 3108M
$ bhosts
HOST_NAME          STATUS       JL/U    MAX  NJOBS    RUN  SSUSP  USUSP    RSV
myhost             ok              -      1      0      0      0      0      0
node1.sas.com      ok              -      1      0      0      0      0      0
node2              ok              -      1      0      0      0      0      0
node3              ok              -      1      0      0      0      0      0
node4              ok              -      1      0      0      0      0      0
$
```

Adding Nodes or SAS Foundation Clients to the Grid

A grid can have machines added to it any time in the future. If a new machine needs to be added to the grid after an initial install, the procedures are similar to adding grid nodes to a new LSF cluster. To add a node to an existing LSF cluster, do the following:

1. Edit the `lsf.cluster.<cluster_name>` file (`lsf.cluster.sas_cluster` in our case) and add the new machine names in the host section. This section looks like the example below:

```
Begin Host
HOSTNAME            model type    server r1m  mem  swp  RESOURCES
#Keywords
myhost               !    LINUX86    1    -    -    -   (linux)
node1.sas.com        !    LINUX86    1    -    -    -   (linux)
node2                !    LINUX86    1    -    -    -   (linux)
node3                !    LINUX86    1    -    -    -   (linux)
node4                !    LINUX86    1    -    -    -   (linux)
End     Host
```

For example, to add **node5** to the previous cluster, the resulting Host section would look like the example below:

```
Begin Host
HOSTNAME            model type    server r1m  mem  swp  RESOURCES
#Keywords
```

```
myhost              !    LINUX86    1    -    -    -   (linux)
node1.sas.com       !    LINUX86    1    -    -    -   (linux)
node2               !    LINUX86    1    -    -    -   (linux)
node3               !    LINUX86    1    -    -    -   (linux)
node4               !    LINUX86    1    -    -    -   (linux)
node5               !    LINUX86    1    -    -    -   (linux)
End     Host
```

2. Follow the steps at the beginning of this chapter.

Converting a Grid Node Machine to a Grid Client

If you are running SAS Foundation to submit jobs to the grid but the machine will not participate as a grid node, and you installed Platform LSF on a machine as an "LSF Server" host type, prevent jobs from running on the machine by making it, in effect, an "LSF Client" machine. Change the state of a machine to 'closed' by following these steps:

1. Log on as the LSF Administrator.
2. Run the command **badmin hclose <*host_name*>**.

When you run the **bhosts** command, the host displays a status of '**closed**'.

Adding a New Machine Type to the Grid

Before adding a new machine type to an existing grid, verify that the host type does not already exist in your cluster by logging onto any host in the cluster and listing the contents of the `LSF_TOP/<version>` directory. If the host type currently exists, a subdirectory with the name of the host type is to display, and it's recommended that you edit the `LSF_CONFDIR/lsf.cluster.<clustername>` file to add the hostname in the HOST section. Then go to step 5 below. If the host type does not already exist, complete all the steps below.

1. Get the LSF distribution tar file for the host type you want to add.
2. Log on as root to any host that can access the LSF install directory.
3. Change to the LSF install directory.
4. Edit `install.config`:
   a. For `LSF_TARDIR`, specify the path to the tar file. For example:
      ```
      LSF_TARDIR="/usr/share/lsf_distrib/9.1"
      ```
   b. For `LSF_ADD_SERVERS`, list the new host names enclosed in quotes and separated by spaces. For example:
      ```
      LSF_ADD_SERVERS="hosta hostb"
      ```
   c. Run `./lsfinstall -f install.config`

   This automatically creates the host information in lsf.cluster.cluster_name.
5. Run `lsadmin reconfig` to reconfigure LIM.
6. Run `badmin reconfig` to reconfigure mbatchd.
7. Run `hostsetup` to set up the new host and configure the daemons to start automatically at boot. For example, from an install directory such as `/usr/share/lsf/9.1/install`:

```
./hostsetup --top="/usr/share/lsf" --boot="y" --profile="y"
--start="y"
```

Note: *There are two dashes "`--` " in the options. See "Chapter 5 – LSF Quick Reference" for more information on the* `hostsetup` *command.*

8. Start LSF on the new host:

```
lsadmin limstartup
lsadmin resstartup
badmin hstartup
```

Setting up UNIX as a LSF Client Installation for Non-Access to a Shared Directory

Use this configuration setup if your machine does not have access to a shared directory.

Use slave.config to install a slave host. Note that slave.config is located in the same directory as install.config after you untar the installer package.

Note: *Seeing the term, "UNAVAILABLE" display during installation is "normal" for a LSF client configuration that is not running LSF daemons.*

1. In slave.config, you need to specify the following params. Even though some params are not needed for client, you need to provide them for the installer to work.

   LSF_TOP -- this should be on a local directory but not the same one as noted for the master host.
   LSF_LICENSE
   LSF_ADMINS -- specify the existing admins of the main cluster.
   LSF_ADD_CLIENTS -- specify the host you are installing on (that is the slave and client host).
   LSF_SERVER_HOSTS -- specify the master and master candidate hosts of the main cluster (in the same order as in the main cluster).
   LSF_LIM_PORT -- specify the LIM port of the main cluster.

   This setup allows you to run "sudo ./lsfinstall **-s** -f slave.config" to install it.

2. Add a line to the lsf .cluster.xxx file of the main cluster. Be sure to indicate the host is type client. For example,

   HOSTNAME model type server RESOURCES #Keywords
   ...
   ib16b06 ! ! 0 ()

   Then run lsadmin reconfig. (badmin reconfig is probably not needed, but was run in this test).

3. On the client host, you can try lsid, and bsub to submit a job. On bsub, you need to specify -R "type==any" as LSF won't detect the host type of a client machine.

   The client host does not refer to to LSF_TOP of the main cluster. The example below shows how lsf.conf looks on the client machine after these steps (from LSF10.1):

LSF_GET_CONF=lim
LSF_CONFDIR=/usr/local/lsf/conf
LSF_LIM_PORT=3789
LSF_SERVER_HOSTS="myMasterHost"
LSF_VERSION=10.1
#LSF_LOCAL_RESOURCES="[hostname hosta][model model1][type type1][server 1][resource resource1][resourcemap n*resource1]"
LSF_TOP=/usr/local/lsf

# Daemon log messages
LSF_LOGDIR=/usr/local/lsf/log
LSF_LOG_MASK=LOG_WARNING
LSF_ENABLE_EGO=N
# LSF_EGO_ENVDIR=/usr/local/lsf/conf
LSB_ENABLE_HPC_ALLOCATION=Y
LSF_EGO_DAEMON_CONTROL=N

4. When running the boot (hostsetup) command for LSF with LSF 9.1, you may receive a message about a missing "LSF Entitlement" in the dialog.

   There are two solutions to correct this issue.
   1. Ignore the problem and let it display an error. Therefore, you do not need an LSF entitlement.
   2. Workaround solution: Is there a bug in the hostsetup command? If so, set the following two environment variables before running the hostsetup script.

   IS_SAS_BUILD=Y
   LSF_LICENSE=<path to the license.dat>

   Example:

   ```
   a. [root@xxxx install]# export IS_SAS_BUILD=Y
   b. [root@xxxx install]# export LSF_LICENSE=/opt/sasinside/LSF_Share
      /license.dat

   c. [root@xxxx install]# ./hostsetup –- top=”/opt/sasinside/LSF_Share
      /lsf" --boot="y" --profile="y" --start="y"
   ```

   Logging installation sequence in /opt/sasinside/LSF_Share/lsf/log/Install.log

Startup and Boot Setup for LSF and PM

LSF Boot Setup with Startup

Change into the LSF_TOP/9*/install subdirectory. For example:

```
[root@localhost] # cd /common/share/lsf/9/install
```

Run the following command initialization files for future reboots after sourcing the LSF profile. This will add the LSF daemons to the server startup and startup the required LSF daemons. Note the two dashes before each parameter.

```
[root@localhost] #  ./hostsetup --top=”/common/share/lsf” --boot=”y”
  --start=”y”
```

PM (jfd) Boot Setup

Set up the Process Manager environment by sourcing the profile.js file if it was not already run. Then, run the PM/jfd “bootsetup” script.

```
[root@localhost] # . <JS_TOP>/conf/profile.js

[root@localhost] # . <JS_TOP>/9*/install/bootsetup
```

Example:

```
[root@localhost] # .  /common/share/pm/conf/profile.js

[root@localhost] # /common/share/pm/9*/install/bootsetup
```

Reminder: (**bootsetup** with no arguments) will install PM/jfd startup at boot time.

Chapter 5 - LSF Quick Reference

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>lsid</td><td>Displays version number, cluster name, and the grid control server (LSF master host) name. Useful to see if the grid daemons are running and if running in SAS mode.</td></tr>
<tr><td>lshosts</td><td>Displays information about the hosts recognized by LSF along with their static resource information.</td></tr>
<tr><td>lsload</td><td>Displays the dynamic resource information for the hosts in the grid (cluster).</td></tr>
<tr><td>bhosts</td><td>Displays batch information about all hosts in the grid (cluster).</td></tr>
<tr><td>bjobs</td><td>Displays information about current user’s LSF jobs</td></tr>
<tr><td>lsfstartup</td><td>Starts the LIM, RES, sbatchd, and mbatchd daemons on all hosts in the cluster. Must be run as root and all hosts must be running rsh or ssh daemons.</td></tr>
<tr><td>lsfrestart</td><td>Restarts the LIM, RES, sbatchd, and mbatchd daemons on all hosts in the cluster. Must be run as root and all hosts must be running rsh or ssh daemons.</td></tr>
<tr><td>lsfshutdown</td><td>Shuts down the LIM, RES, sbatchd, and mbatchd daemons daemons on all hosts in the cluster. Must be run as root and all hosts must be running rsh or ssh daemons.</td></tr>
<tr><td>lsadmin</td><td>Administrative tool for LSF available to LSF administrators. Useful subcommands include the following:
<table>
<tr><td>reconfig</td><td>Restarts all LIMs in the cluster to read any changes in the configuration files.</td></tr>
<tr><td>limstartup</td><td>Starts LIM on the local host</td></tr>
<tr><td>limrestart</td><td>Restarts LIM on the local host</td></tr>
<tr><td>resstartup</td><td>Starts RES on local host</td></tr>
<tr><td>resrestart</td><td>Restarts RES on local host</td></tr>
</table></td></tr>
<tr><td>bhist</td><td>Displays historical information about jobs. Useful parameters including the following:
<table>
<tr><td>-p | -r | -d | -a</td><td>Displays information about specific jobs (<u>p</u>ending, <u>r</u>unning, <u>d</u>one, or <u>a</u>ll).</td></tr>
<tr><td>-l</td><td>Display in long format.</td></tr>
<tr><td>-u <user> | all</td><td>Displays job for specified or all users.</td></tr>
<tr><td><job ID></td><td>Displays only specified job information.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>badmin</td><td>Administrative tool for LSF’s batch processing facility available to LSF administrators. Useful subcommands include the following:
<table>
<tr><td>reconfig</td><td>Reconfigures the batch facility without restarting sbatchd or mbatchd to read any changes in the configuration files.</td></tr>
<tr><td>hstartup</td><td>Starts sbatchd on the local host</td></tr>
<tr><td>hrestart</td><td>Restarts sbatchd on the local host</td></tr>
<tr><td>mbdrestart</td><td>Restarts mbatchd. Needs to be done when new hosts are added to the grid (cluster).</td></tr>
<tr><td>hclose <host></td><td>Closes a host preventing it from running jobs.</td></tr>
<tr><td>hopen <host></td><td>Opens a host to allow it to run jobs.</td></tr>
</table></td></tr>
<tr><td>bsub</td><td>Submit a job to the grid. Useful parameters include the following:
<table>
<tr><td>-I</td><td>Interactive. Remote output displayed locally.</td></tr>
<tr><td>-m</td><td>Submit to a specific host.</td></tr>
<tr><td>-R “res_req”</td><td>Submit with specified resource.</td></tr>
</table></td></tr>
<tr><td>hostsetup</td><td>Sets up a host to use the LSF cluster and configures LSF daemons to start automatically.
<table>
<tr><td>--top</td><td>Top-level installation directory that contains the cluster the local host belongs to. <path> must be accessible to local host where hostsetup is running</td></tr>
<tr><td>--boot</td><td>Configure system scripts to automatically start and stop LSF at system startup and shutdown. The local host where hostsetup is running must be an LSF server in the cluster.</td></tr>
<tr><td>--profile</td><td>Add cshrc.lsf and profile.lsf to system-wide environment and startup programs.</td></tr>
<tr><td>--start</td><td>Start LSF on the local host after hostsetup. The local host where hostsetup is running must be an LSF server in the cluster.</td></tr>
<tr><td>--quiet</td><td>Do not display detailed messages.</td></tr>
</table></td></tr>
</table>

The LSF commands shown in this section include examples of typical output. The output you see differs according to your local configuration.

The commands are described briefly so that you can easily use them as a “confirmation check” for your LSF installation. See the *LSF Reference* for complete usage and command options. You can use these commands on any LSF host. If you get proper output from these commands, your cluster is ready to use. If your output from the commands discussed in this section has errors, see the *LSF Reference* for help.

Check Cluster Configuration (lsadmin)

`lsadmin ckconfig -v`

The **lsadmin** command controls the operation of an LSF cluster and LSF configuration files. The -v flag displays detailed information about the LSF configuration:

```
$ lsadmin ckconfig -v


Checking configuration files ...

Platform EGO 1.2.10.0 build 243073, Feb 03 2015
Copyright IBM Corp. 1992, 2014. All rights reserved.
US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.

  binary type: linux2.6-glibc2.3-x86_64
Reading configuration from
/usr/share/lsf/conf/ego/lax94t01_pss91/kernel/ego.conf
Jun 15 13:44:39 2015 9687 5 1.2.10 Platform EGO 1.2.10.0 build 243073,
Feb 03 2015
Copyright IBM Corp. 1992, 2014. All rights reserved.
US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.

  binary type: linux2.6-glibc2.3-x86_64

Jun 15 13:44:39 2015 9687 6 1.2.10 Lim starting...
Jun 15 13:44:39 2015 9687 6 1.2.10 LIM is running in advanced workload
execution mode.
Jun 15 13:44:39 2015 9687 6 1.2.10 Master LIM is not running in
EGO_DISABLE_UNRESOLVABLE_HOST mode.
Jun 15 13:44:39 2015 9687 5 1.2.10 /usr/share/lsf/9.1/linux2.6-glibc2.3-
x86_64/etc/lim -C
Jun 15 13:44:39 2015 9687 6 1.2.10 initEntitlement: EGO_AUDIT_MAX_SIZE
was not set. Default value <100> will be used.
Jun 15 13:44:39 2015 9687 6 1.2.10 initEntitlement: EGO_AUDIT_MAX_ROTATE
was not set. Default value <20> will be used.
Jun 15 13:44:39 2015 9687 6 1.2.10 LIM is running as IBM Platform LSF
Standard Edition.
Jun 15 13:44:39 2015 9687 3 1.2.10 domanager():
/usr/share/lsf/conf/lsf.cluster.lax94t01_pss91(15): The cluster manager
is the invoker <sasidb2> in debug mode
Jun 15 13:44:39 2015 9687 6 1.2.10 reCheckClass: numhosts 5 so reset
exchIntvl to 15.00
Jun 15 13:44:39 2015 9687 6 1.2.10 Checking Done.
------------------------------------------------------------

No errors found.
```

The messages shown are typical of normal output from **lsadmin ckconfig -v**. Other messages may indicate problems with your LSF configuration. See the *LSF Reference* for help with some common configuration errors.

Find Out Cluster Status (lsid and lsload)

lsid

Informs you if your LSF environment is set up properly. Cluster status lsid displays the current LSF version number, cluster name, and host name of the current grid control server (LSF master host) for your cluster. The grid control server (LSF master) name displayed by `lsid` may vary, but it is usually the first host configured in the Hosts section of `LSF_CONFDIR/lsf.cluster.`*`cluster_name`*.

lsid

```
IBM Platform LSF Standard 9.1.3.0, Feb 03 2015
Copyright IBM Corp. 1992, 2014. All rights reserved.
US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

My cluster name is sas_cluster
My master name is myhost
Cluster in ISV mode: SAS
```

If you see the message

```
Cannot open lsf.conf file
```

the LSF_ENVDIR environment variable is probably not set correctly. Use `cshrc.lsf` or `profile.lsf` to set up your environment.

lsload

Displays the current load levels of the cluster. The output contains one line for each host in the cluster. The status is correct for all hosts in your cluster. For example:

```
lsload


HOST_NAME status r15s  r1m r15m  ut   pg  ls   it  tmp  swp   mem
hosta         ok  0.0  0.0  0.0  6%  0.2   2 1365  97M  65M   29M
hostb         ok  0.0  0.0  0.0  9%  0.0   4    1 130M 319M   12M
hostc         ok  2.5  2.2  1.9 64% 56.7  50    0 929M 931M 4000M
hostd         ok  0.2  0.2  0.2  1%  0.0   0  367  93M  86M   50M
hoste       busy *6.0  2.2  1.9 64% 56.7  50    0 929M 931M 4000M
hostf    unavail
```

A `busy` status is shown for hosts with any load index beyond its configured thresholds. An asterisk (`*`) marks load indices that are beyond their thresholds, causing the host status to be `busy`. A minus sign (`-`) in front of the value `ok` means that RES is not running on that host.

If you see the message

```
LIM is down
```

or

```
LIM is not responding
```

after starting or reconfiguring LSF, wait a few seconds and try `lsload` again to give the LIMs time to initialize. Notice that `lsload` also shows if LSF is licensed for the host. If you see the message

```
Host does not have a software license
```

you must install a valid LSF license or make sure that the license server is running properly.

There are also a couple of other useful commands:

- The `lshosts` command displays configuration information for LSF hosts and their static resource information.
- The `lsinfo` command displays cluster configuration information about resources, host types, and host models.

Check LSF Batch Configuration (badmin)

`badmin ckconfig –v`

The `badmin` command controls and monitors the operation of the LSF Batch system. Use the `badmin ckconfig` command to check the LSF Batch configuration files. The `-v` flag displays detailed information about the configuration:

```
badmin ckconfig –v
Checking configuration files ...
-------------------------------------------------------
No errors found.
```

The messages shown above are the normal output from `badmin ckconfig -v`. Other messages may indicate problems with the Platform LSF Batch configuration. See the *LSF Reference* for help with some common configuration errors.

Find Out LSF Batch System Status (bhosts and bqueues)

`bhosts`

The `bhosts` command tells you if LSF Batch is running properly. Notice that `Bhosts` displays the status and other details about the grid nodes (LSF Batch server hosts) in the cluster:

- maximum number of job slots allowed by a single user
- total number of jobs in the system, jobs running, jobs suspended by users, and jobs suspended by the system
- total number of reserved job slots

The status is designed to be `ok` for all grid nodes (hosts) in your cluster. For example:

```
bhosts


HOST_NAME STATUS JL/U MAX NJOBS RUN SSUSP USUSP RSV
hosta ok            -   -     0   0     0     0   0
hostb ok            -   -     0   0     0     0   0
hostc ok            -   -     0   0     0     0   0
hostd ok            -   -     0   0     0     0   0
```

If you see the message

```
lsbatch daemons not responding
```

after starting or reconfiguring LSF, wait a few seconds and try `bhosts` again to give the SBDs time to initialize.

bqueues

LSF Batch queues organize jobs with different priorities and different scheduling policies. The `bqueues` command displays available queues and their configuration parameters. For a queue to accept and dispatch jobs, the status is to be `Open:Active`.

bqueues

```
QUEUE_NAME      PRIO STATUS          MAX JL/U JL/P JL/H NJOBS  PEND   RUN  SUSP
owners           43  Open:Active       -    -    -    -     0     0     0     0
priority         43  Open:Active       -    -    -    -     0     0     0     0
night            40  Open:Inact        -    -    -    -     0     0     0     0
chkpnt_rerun_qu  40  Open:Active       -    -    -    -     0     0     0     0
short            35  Open:Active       -    -    -    -     0     0     0     0
license          33  Open:Active       -    -    -    -     0     0     0     0
normal           30  Open:Active       -    -    -    -     0     0     0     0
hpc_linux        30  Open:Active       -    -    -    -     0     0     0     0
hpc_linux_tv     30  Open:Active       -    -    -    -     0     0     0     0
unicodecmd       30  Open:Active       -    -    -    -     0     0     0     0
idle             20  Open:Active       -    -    -    -     0     0     0     0
```

The queue information displayed by `bqueues` is configured in `lsb.queues`. Eight queues are defined by default in `lsb.queues`. Modify this file to add, delete, or change queues.

bqueues -l

To see more detailed queue information, use `bqueues -l`:

```
bqueues -l normal
```

```
QUEUE: normal
  -- For normal low priority jobs, running only if hosts are lightly loaded.
This is the default queue.

PARAMETERS/STATISTICS
PRIO NICE STATUS          MAX JL/U JL/P JL/H NJOBS  PEND   RUN SSUSP USUSP  RSV
 30   20  Open:Active       -    -    -    -     0     0     0     0     0    0

SCHEDULING PARAMETERS
           r15s   r1m  r15m   ut      pg    io   ls    it    tmp    swp    mem
 loadSched   -     -     -     -       -     -    -     -     -      -      -
 loadStop    -     -     -     -       -     -    -     -     -      -      -

USERS: all
HOSTS:  all
```

`bqueues -l` shows the following kinds of information about the queue:

- What kinds of jobs are meant to run on the queue?
- Resource usage limits.
- Nodes (hosts) and users that are able to use the queue.
- Scheduling threshold values:
  - `loadSched` is the threshold for LSF to dispatch a job automatically.
  - `loadStop` is the threshold for LSF to suspend a job automatically.

There are a couple of other useful commands:

- The `bparams` command displays information about the LSF Batch configuration parameters.
- The `bhist` command displays historical information about jobs.

Verifying the Network Setup

Overview

The first step in troubleshooting problems with a SAS grid is to verify that all computers in the grid can communicate with one another through the ports that are used by the grid middleware.

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 signon connection to the grid node to fail or to hang. This happens because the SAS session 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 in the grid. Follow these steps to test the network setup:

1. Run the `hostname` command on every machine in 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 in 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 in 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 in 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 to verify that communication works between the LSF daemons and the instances of SAS. Issue the `telnet` *<host><port>* command to determine whether a port is open on a specific host.

The default ports used in a grid are:

- LSF: 6878, 6881, 6882, 7869, 7870, 7871, and 7872
- Grid Monitoring Service: 1976
- 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`
- 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 the `-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
```

For More Information

See the *LSF Administrator's Guide* for more information about seeing the status of your cluster.

See the *LSF Reference* for detailed information about the commands described in this section.

See *Administering Process Manager* for detailed information about Process Manager configuration and maintenance.

These documents are also available at:
http://support.sas.com/rnd/scalability/platform/index.html.

Appendix – LDAP/PAM Authentication

Configure LDAP/PAM Authentication for Process Manager Server

On UNIX systems, Process Manager supports LDAP/PAM authentication through PAM (pluggable authentication modules). PAM is a third-party tool that can be configured to use the `pam_ldap` module from the `libpam-ldap` package to log into the LDAP/PAM server for password checking.

To enable LDAP/PAM authentication for Process Manager, perform the following steps:

1. Set `JS_LOGIN_REQUIRED=true` in `js.conf`.
2. Modify the PAM configuration on your system to add a service name `eauth_userpass` for the module type authorization.

   For example, on Linux, create a new file `eauth_userpass` under the `/etc/pam.d` directory, and then add the following entry to the file:
   ```
   auth required /lib/security/$ISA/pam_ldap.so
   ```

   On Solaris, modify `/etc/pam.conf` to add the following entries:

   ```
   eauth_userpass auth requisite /usr/lib/security/64/pam_authtok_get.so.1
   eauth_userpass auth required /usr/lib/security/64/pam_dhkeys.so.1
   eauth_userpass auth required /usr/lib/security/64/pam_unix_cred.so.1
   eauth_userpass auth binding /usr/lib/security/64/pam_passwd_auth.so.1
   server_policy
   eauth_userpass auth required /usr/lib/security/64/pam_ldap.so.1
   ```

   On AIX, modify `/etc/pam.conf` to add the following entry:

   ```
   eauth_userpass auth required /usr/lib/security/64/pam_aix
   ```
   Note that the absolute path for the `pam_ldap` module may be different on your system. Ensure that you specify the 64-bit `pam_ldap` module on 64-bit operating systems.

3. Restart Process Manager server `jfd`.

Configure LDAP/PAM Authentication for GMS

On UNIX systems, GMS supports LDAP/PAM authentication through pluggable authentication modules (PAM). PAM is a third-party tool that can be configured to use the `pam_ldap` module from the `libpam-ldap` package to log into the LDAP/PAM server for password checking.

To enable LDAP/PAM authentication for GMS, perform the following steps:

1. Set `GA_PAM_ENABLE=Y` in `ga.conf`.
2. Modify the PAM configuration on your system. You have two options for this step:

   Option 1: Modify the existing PAM module: passwd

   For example, on Linux, make sure `/etc/pam.d/passwd` has the following entry:
   ```
   auth required pam_stack.so service=system-auth
   ```

Then modify `/etc/pam.d/system-auth` to add the following:

```
auth sufficient /lib/security/$ISA/pam_ldap.so use_first_pass
```

On Solaris, modify `/etc/pam.conf` to add the following:

```
passwd auth requisite /usr/lib/security/64/pam_authtok_get.so.1
passwd auth required /usr/lib/security/64/pam_dhkeys.so.1
passwd auth required /usr/lib/security/64/pam_unix_cred.so.1
passwd auth binding /usr/lib/security/64/pam_passwd_auth.so.1
server_policy
passwd auth required /usr/lib/security/64/pam_ldap.so.1
```

On AIX, modify `/etc/pam.conf` to add the following:

```
passwd auth required /usr/lib/security/64/pam_aix
```

Option 2: Configure a new `PAM` module specified by `GA_PAM_SERVICE` in `ga.conf`

For example, in `ga.conf`, set the following:

```
GA_PAM_ENABLE=Y
GA_PAM_SERVICE=ga_auth
```

On Linux, create a new file ga_auth under /etc/pam.d directory, and add the following entry to the file:

```
auth required /lib/security/$ISA/pam_ldap.so
```

On Solaris, modify `/etc/pam.conf` to add the following entries:

```
ga_auth auth requisite /usr/lib/security/64/pam_authtok_get.so.1
ga_auth auth required /usr/lib/security/64/pam_dhkeys.so.1
ga_auth auth required /usr/lib/security/64/pam_unix_cred.so.1
ga_auth auth binding /usr/lib/security/64/pam_passwd_auth.so.1
server_policy
ga_auth auth required /usr/lib/security/64/pam_ldap.so.1
```

On AIX, modify `/etc/pam.conf` to add the following entry:

```
ga_auth auth required /usr/lib/security/64/pam_aix
```

Note that the absolute path for the `pam_ldap` module may be different on your system. Make sure you specify the 64-bit `pam_ldap` module on 64-bit operating systems.

Option 1 is recommended. Use option 2 if your site does not allow changing of the passwd module.

3. Restart GMS daemon gabd.

SAS is the leader in business analytics software and services, and the largest independent vendor in the business intelligence market. Through innovative solutions, SAS helps customers at more than 70,000 sites improve performance and deliver value by making better decisions faster. Since 1976 SAS has been giving customers around the world THE POWER TO KNOW®.