
Platform **LSF**[®] Administrator's Primer

Version 6.2

December 2005

Comments to: doc@platform.com



Copyright © 1994-2005 Platform Computing Corporation

All rights reserved.

We'd like to hear from you You can help us make this document better by telling us what you think of the content, organization, and usefulness of the information. If you find an error, or just want to make a suggestion for improving this document, please address your comments to doc@platform.com.

Your comments should pertain only to Platform documentation. For product support, contact support@platform.com.

Although the information in this document has been carefully reviewed, Platform Computing Corporation ("Platform") does not warrant it to be free of errors or omissions. Platform reserves the right to make corrections, updates, revisions or changes to the information in this document.

UNLESS OTHERWISE EXPRESSLY STATED BY PLATFORM, THE PROGRAM DESCRIBED IN THIS DOCUMENT IS PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL PLATFORM COMPUTING BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION ANY LOST PROFITS, DATA, OR SAVINGS, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS PROGRAM.

Document redistribution and translation

This document is protected by copyright and you may not redistribute or translate it into another language, in part or in whole.

Internal redistribution **You may only redistribute this document internally within your organization (for example, on an intranet) provided that you continue to check the Platform Web site for updates and update your version of the documentation. You may not make it available to your organization over the Internet.**

Trademarks ® LSF is a registered trademark of Platform Computing Corporation in the United States and in other jurisdictions.

™ PLATFORM COMPUTING, PLATFORM SYMPHONY, PLATFORM JOBSCHEDULER, PLATFORM ENTERPRISE GRID ORCHESTRATOR, PLATFORM EGO, PLATFORM VM ORCHESTRATOR, PLATFORM VMO, ACCELERATING INTELLIGENCE, and the PLATFORM and PLATFORM LSF logos are trademarks of Platform Computing Corporation in the United States and in other jurisdictions.

UNIX® is a registered trademark of The Open Group in the United States and in other jurisdictions.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

Windows® is a registered trademark of Microsoft Corporation in the United States and other countries.

Macrovision®, Globetrotter®, FLEX/m®, FLEXnet™, FLEXnet Manager™, FLEXnet Connector™, are registered trademarks or trademarks of Macrovision Corporation in the United States of America and/or other countries.

Topspin® is a registered trademark of Topspin Communications, Inc.

Intel®, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Other products or services mentioned in this document are identified by the trademarks or service marks of their respective owners.

Third Party License Agreements

www.platform.com/Company/third.part.license.htm

Third Party Copyright Notices

www.platform.com/Company/Third.Party.Copyright.htm

Contents

Welcome	5
About This Guide	6
Learn About Platform LSF	8
Get Technical Support	9
1 About Your Cluster	11
Cluster Characteristics	12
LSF Directories and Configuration Files	13
2 Working with LSF	17
Starting, Stopping, and Reconfiguring LSF	18
Checking LSF Status	21
Running LSF Jobs	26
3 Managing Your Cluster	31
Managing Users, Hosts, and Queues	32
Configuring LSF Startup	38
4 Working with LSF Licenses	39
About LSF Licenses	40
Setting up a Permanent LSF License	42
5 Using Shared Resources to Manage Software Licenses	47
Managing Software Licenses and Other Shared Resources	48
6 Troubleshooting LSF Problems	53
Common LSF Problems	54
Index	63

Welcome

- Contents
- ◆ “About This Guide” on page 6
 - ◆ “Learn About Platform LSF” on page 8
 - ◆ “Get Technical Support” on page 9

About Platform Computing

Platform Computing is the largest independent grid software developer, delivering intelligent, practical enterprise grid software and services that allow organizations to plan, build, run and manage grids by optimizing IT resources. Through our proven process and methodology, we link IT to core business objectives, and help our customers improve service levels, reduce costs and improve business performance.

With industry-leading partnerships and a strong commitment to standards, we are at the forefront of grid software development, propelling over 1,600 clients toward powerful insights that create real, tangible business value. Recognized worldwide for our grid computing expertise, Platform has the industry's most comprehensive desktop-to-supercomputer grid software solutions. For more than a decade, the world's largest and most innovative companies have trusted Platform Computing's unique blend of technology and people to plan, build, run and manage grids.

Learn more at www.platform.com.

About This Guide

Last update December 20 2005

Latest version www.platform.com/Support/Documentation.htm

Purpose of this guide

This guide is your starting point for learning how to manage and use your new cluster running the Platform **LSF**[®] software (“LSF”). It provides an overview of LSF concepts, basic commands to test your new cluster, how to run applications through LSF, LSF licensing, and some troubleshooting tips.

Who should use this guide

This guide is written for new Platform LSF cluster administrators who want to familiarize themselves with the fundamentals of managing and using an LSF cluster running on UNIX systems.

What you should already know

This guide assumes you have already followed the installation procedure described in *Installing Platform LSF on UNIX and Linux*, and performed the the steps for host setup in the `lsf_getting_started.html` file generated by `lsfinstall`.

To understand and use LSF, you should be familiar with common UNIX system administration tasks such as creating user accounts, sharing and mounting Network File System (NFS) partitions, and backing up the system

How to find out more

To learn more about LSF:

- ◆ See *Administering Platform LSF* for detailed information about LSF concepts and tasks.
- ◆ See the *Platform LSF Reference* for detailed information about LSF commands, files, and configuration parameters.
- ◆ See “[Learn About Platform LSF](#)” on page 8 for additional resources.

Typographical conventions

Typeface	Meaning	Example
Courier	The names of on-screen computer output, commands, files, and directories	The <code>lsid</code> command
Bold Courier	What you type, exactly as shown	Type <code>cd /bin</code>
<i>Italics</i>	<ul style="list-style-type: none"> ◆ Book titles, new words or terms, or words to be emphasized ◆ Command-line place holders—replace with a real name or value 	The queue specified by <i>queue_name</i>
Bold Sans Serif	<ul style="list-style-type: none"> ◆ Names of GUI elements that you manipulate 	Click OK

Command notation

Notation	Meaning	Example
Quotes " or '	Must be entered exactly as shown	<code>"job_ID[index_list]"</code>
Commas ,	Must be entered exactly as shown	<code>-C time0,time1</code>

Notation	Meaning	Example
Ellipsis ...	The argument before the ellipsis can be repeated. Do not enter the ellipsis.	<i>job_ID</i> ...
lower case italics	The argument must be replaced with a real value you provide.	<i>job_ID</i>
OR bar	You must enter one of the items separated by the bar. You cannot enter more than one item, Do not enter the bar.	[-h -V]
Parenthesis ()	Must be entered exactly as shown	-X " <i>exception_cond</i> ([<i>params</i>]): <i>action</i>] ...
Option or variable in square brackets []	The argument within the brackets is optional. Do not enter the brackets.	lsid [-h]
Shell prompts	<ul style="list-style-type: none"> ◆ C shell: % ◆ Bourne shell and Korn shell: \$ ◆ root account: # Unless otherwise noted, the C shell prompt is used in all command examples	% cd /bin

Learn About Platform LSF

World Wide Web and FTP

The latest information about all supported releases of Platform LSF is available on the Platform Web site at www.platform.com. Look in the Online Support area for current README files, Release Notes, Upgrade Notices, Frequently Asked Questions (FAQs), Troubleshooting, and other helpful information.

The Platform FTP site ([ftp.platform.com](ftp://platform.com)) also provides current README files, Release Notes, and Upgrade information for all supported releases of Platform LSF.

Visit the Platform User Forum at www.platformusers.net to discuss workload management and strategies pertaining to distributed and Grid Computing.

If you have problems accessing the Platform web site or the Platform FTP site, contact support@platform.com.

Platform training

Platform's Professional Services training courses can help you gain the skills necessary to effectively install, configure and manage your Platform products. Courses are available for both new and experienced users and administrators at our corporate headquarters and Platform locations worldwide.

Customized on-site course delivery is also available.

Find out more about [Platform Training](http://www.platform.com/training) at www.platform.com/training, or contact Training@platform.com for details.

README files, release notes and UPGRADE

Before installing LSF, be sure to read the files named `readme.html` and `release_notes.html`. To upgrade to Version 6.2, follow the steps in `upgrade.html`.

You can also view these files from the Download area of the Platform Online Support Web page.

Platform documentation

Documentation for Platform products is available in HTML and PDF format on the Platform Web site at

<http://www.platform.com/Support/Documentation.htm>.

Get Technical Support

Contact Platform

Contact Platform Computing or your LSF vendor for technical support. Use one of the following to contact Platform technical support:

Email support@platform.com

World Wide Web www.platform.com

Mail Platform Support
Platform Computing Corporation
3760 14th Avenue
Markham, Ontario
Canada L3R 3T7

When contacting Platform, please include the full name of your company.

See the Platform Web site at www.platform.com/contactus for other contact information.

We'd like to hear from you

If you find an error in any Platform documentation, or you have a suggestion for improving it, please let us know:

Email doc@platform.com

Mail Information Development
Platform Computing Corporation
3760 14th Avenue
Markham, Ontario
Canada L3R 3T7

Be sure to tell us:

- ◆ The title of the manual you are commenting on
- ◆ The version of the product you are using
- ◆ The format of the manual (HTML or PDF)

Where to go next

Familiarize yourself with your new Platform LSF cluster, described in Chapter 1, “[About Your Cluster](#)”.

About Your Cluster

- Contents**
- ◆ [“Cluster Characteristics”](#) on page 12
 - ◆ [“LSF Directories and Configuration Files”](#) on page 13

Cluster Characteristics

- Contents
- ◆ “Cluster name and administrators” on page 12
 - ◆ “LSF hosts” on page 12

Cluster name and administrators

The cluster name you specified at installation is part of the name of `LSF_CONFDIR/lsf.cluster.cluster_name`. For example:

```
/usr/share/lsf/lsf_62/conf/lsf.cluster.lsf_62
```

Cluster administrators are listed in the `ClusterAdmins` section of `LSF_CONFDIR/lsf.cluster.cluster_name`.

LSF hosts

- ◆ Host types installed in your cluster are listed in the `Hosts` section of `LSF_CONFDIR/lsf.cluster.cluster_name`.
- ◆ The LSF master host is the first host configured in the `Hosts` section of `LSF_CONFDIR/lsf.cluster.cluster_name`.
- ◆ LSF server hosts defined in your cluster are indicated by 1 in the `server` column of the `Hosts` section of `LSF_CONFDIR/lsf.cluster.cluster_name`.
- ◆ LSF client-only hosts defined in your cluster are indicated by 0 (zero) in the `server` column in `LSF_CONFDIR/lsf.cluster.cluster_name`.

LSF Directories and Configuration Files

- Contents
- ◆ “Four important LSF configuration files” on page 13
 - ◆ “Default directory structure” on page 13
 - ◆ “LSF directories” on page 14
 - ◆ “LSF cluster configuration files” on page 15
 - ◆ “LSF Batch configuration files” on page 15
 - ◆ “Daemon log files” on page 16

Four important LSF configuration files

LSF configuration administered through several configuration files, which you use to modify the behavior of your cluster. The four most important files you will work with most often are:

- ◆ `LSF_CONFDIR/lsf.conf`
- ◆ `LSF_CONFDIR/lsf.cluster.cluster_name`
- ◆ `LSF_CONFDIR/lsf.shared`
- ◆ `LSB_CONFDIR/cluster_name/configdir/lsb.queues`

These files are created during LSF installation according to the options you specified in the `install.config` file. After installation, you can change the configuration parameters in these files to suit the needs of your site.

Who owns these files Except for `LSF_CONFDIR/lsf.conf`, which is owned by root, all of these files are owned by the primary LSF administrator, and readable by all cluster users.

lsf.conf The most important file in LSF. It is owned by root and contains the paths to the LSF configuration directories, log directories, libraries, license file and other global configuration information. The location of `lsf.conf` is defined by the `LSF_ENVDIR` variable. If LSF cannot find this file, it cannot start properly.

By default, LSF checks the directory defined by `LSF_ENVDIR` for `lsf.conf`. If `lsf.conf` is not in `LSF_ENVDIR`, LSF looks for it in `/etc`.

lsf.cluster.cluster_name

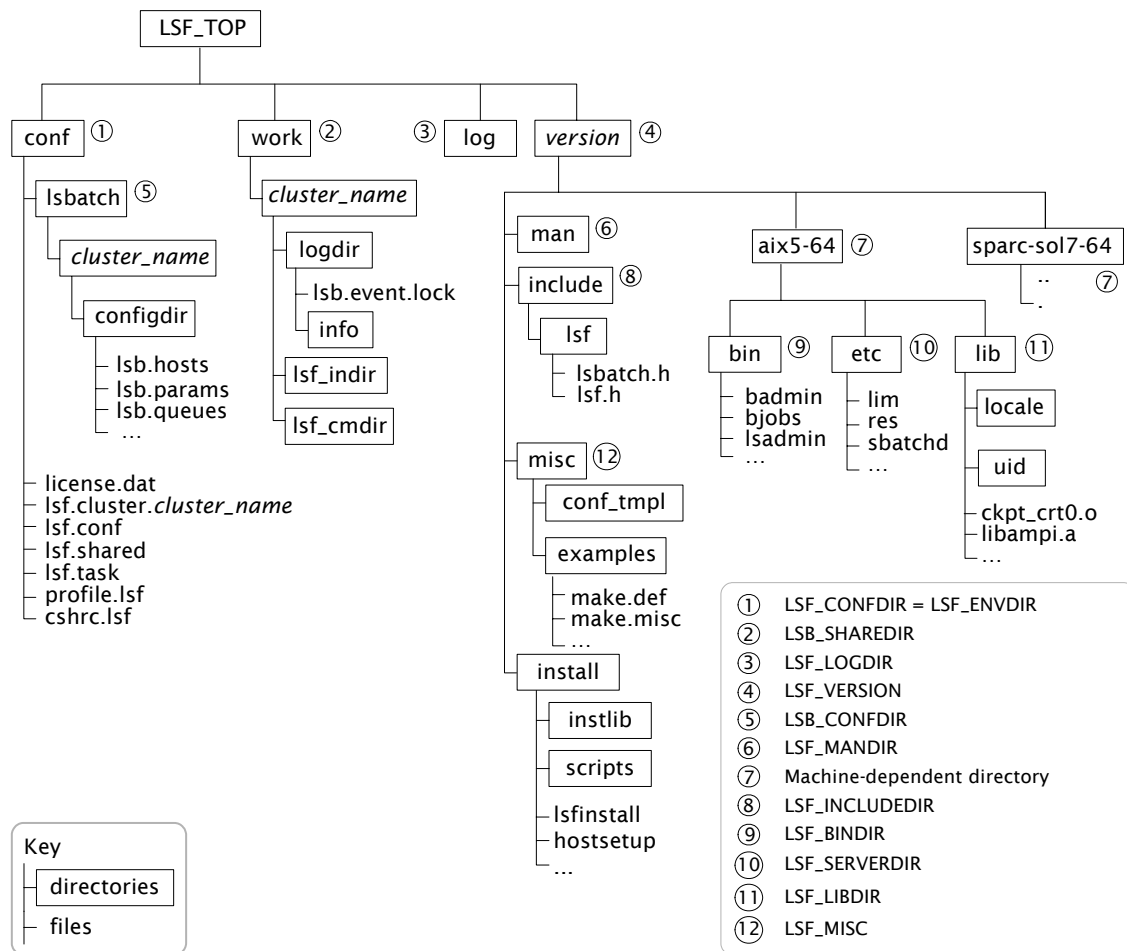
Defines the host names, models, and types of all of the hosts in the LSF cluster. It also defines the user names of the LSF administrators, and the locations of different shared resources for one LSF cluster.

lsf.shared This file is like a dictionary that defines all the keywords used by the LSF cluster. You can add your own keywords to specify the names of resources or host types.

lsb.queues Defines the LSF batch queues and their parameters for one LSF cluster.

Default directory structure

The following diagram illustrates a typical directory structure for an example installation. Depending on which products you have installed and platforms you have selected, your directory structure may vary.



LSF directories

The following directories are owned by the primary LSF administrator and are readable by all cluster users:

Directory	Description	Example
LSF_CONFDIR	LSF configuration directory	/usr/share/lsf/lsf_62/conf/
LSB_CONFDIR	LSF batch configuration directory	/usr/share/lsf/lsf_62/conf/lsbatch/
LSB_SHAREDIR	LSF batch job history directory	/usr/share/lsf/lsf_62/work/
LSF_LOGDIR	Server daemon error logs, one for each LSF daemon	/usr/share/lsf/lsf_62/log/

The following directories are owned by root and are readable by all cluster users:

Directory	Description	Example
LSF_BINDIR	LSF user commands, shared by all hosts of the same type	/usr/share/lsf/lsf_62/6.2/ sparc-sol2/bin/
LSF_INCLUDEDIR	Header files <code>lsf/lsf.h</code> and <code>lsf/lsbatch.h</code>	/usr/share/lsf/lsf_62/6.2/include/
LSF_LIBDIR	LSF libraries, shared by all hosts of the same type	/usr/share/lsf/lsf_62/6.2/ sparc-sol2/lib/

Directory	Description	Example
LSF_MANDIR	LSF man pages	<code>/usr/share/lsf/lsf_62/6.2/man/</code>
LSF_MISC	Examples and other miscellaneous files	<code>/usr/share/lsf/lsf_62/6.2/misc/</code>
LSF_SERVERDIR	Server daemon binaries, scripts and other utilities, shared by all hosts of the same type	<code>/usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/</code>
LSF_TOP	Top-level installation directory	<code>/usr/share/lsf/lsf_62/</code>

Other configuration directories are specified in `LSF_CONFDIR/lsf.conf`.

LSF cluster configuration files

The following files are owned by the primary LSF administrator and are readable by all cluster users:

File	Example
License file The LSF license file is usually installed in <code>LSF_CONFDIR</code> , and defined by <code>LSF_LICENSE_FILE</code> in <code>lsf.conf</code> .	<code>/usr/share/lsf/lsf_62/conf/license.dat</code>
Global configuration file describing the configuration and operation of the cluster	<code>/usr/share/lsf/lsf_62/conf/lsf.conf</code>
Keyword definition file shared by all clusters. Defines cluster name, host types, host models, and site-specific resources	<code>/usr/share/lsf/lsf_62/conf/lsf.shared</code>
Cluster configuration file that defines hosts, administrators, and location of site-defined shared resources	<code>/usr/share/lsf/lsf_62/conf/lsf.cluster.cluster_name</code>
Mapping file for task names and their default resource requirements	<code>/usr/share/lsf/lsf_62/conf/lsf.task</code> <code>/usr/share/lsf/lsf_62/conf/lsf.task.cluster_name</code>

LSF Batch configuration files

The following files are owned by the primary LSF administrator and are readable by all cluster users:

File	Example
Server hosts and their attributes, such as scheduling load thresholds, dispatch windows, and job slot limits. If no hosts are defined in this file, then all LSF server hosts listed in <code>LSF_CONFDIR/lsf.cluster.cluster_name</code> are assumed to be LSF Batch server hosts.	<code>/usr/share/lsf/lsf_62/conf/lsbatch/lsf_62/configdir/lsb.hosts</code>
LSF scheduler and resource broker plugin modules. If no scheduler or resource broker modules are configured, LSF uses the default scheduler plugin module named <code>schmod_default</code> .	<code>/usr/share/lsf/lsf_62/conf/lsbatch/lsf_62/configdir/lsb.modules</code>
LSF Batch system parameter file	<code>/usr/share/lsf/lsf_62/conf/lsbatch/lsf_62/configdir/lsb.params</code>
Job queue definitions	<code>/usr/share/lsf/lsf_62/conf/lsbatch/lsf_62/configdir/lsb.queue</code>
Resource allocation limits, exports, and resource usage limits.	<code>/usr/share/lsf/lsf_62/conf/lsbatch/lsf_62/configdir/lsb.resources</code>
LSF user groups, hierarchical fairshare for users and user groups, and job slot limits for users and user groups. Also used to configure account mappings in a MultiCluster environment.	<code>/usr/share/lsf/lsf_62/conf/lsbatch/lsf_62/configdir/lsb.users</code>

Daemon log files

LSF server daemon log files are stored in the directory specified by `LSF_LOGDIR` in `LSF_CONFDIR/lsf.conf`.

File	Example
Load Information Manager (<code>lim</code>)	<code>/usr/share/lsf/lsf_62/log/lim.log.hosta</code>
Remote Execution Server (<code>res</code>)	<code>/usr/share/lsf/lsf_62/log/res.log.hosta</code>
Master Batch Daemon (<code>mbatchd</code>)	<code>/usr/share/lsf/lsf_62/log/mbatchd.log.hosta</code>
Master Scheduler Daemon (<code>mbschd</code>)	<code>/usr/share/lsf/lsf_62/log/mbschd.log.hosta</code>
Slave Batch Daemon (<code>sbatchd</code>)	<code>/usr/share/lsf/lsf_62/log/sbatchd.log.hosta</code>
Process Information Manager (<code>pim</code>)	<code>/usr/share/lsf/lsf_62/log/pim.log.hosta</code>

Who owns and who should write to this directory Make sure that the primary LSF administrator owns the LSF log directory (`LSF_LOGDIR`), and that `root` can write to this directory. If an LSF server is unable to write to `LSF_LOGDIR`, then the error logs are created in `/tmp`.

For more information

- ◆ See *Administering Platform LSF* for information about cluster configuration files and parameters.
- ◆ See the *Platform LSF Reference* for more information about cluster configuration files and parameters.

Where to go next

Start using your new Platform LSF cluster, described in Chapter 2, “[Working with LSF](#)”.

Working with LSF

- Contents**
- ◆ “Starting, Stopping, and Reconfiguring LSF” on page 18
 - ◆ “Checking LSF Status” on page 21
 - ◆ “Running LSF Jobs” on page 26

Starting, Stopping, and Reconfiguring LSF

- Contents
- ◆ “Two LSF administration commands (`lsfadmin` and `badadmin`)” on page 18
 - ◆ “Setting up the LSF environment (`cshrc.lsf` and `profile.lsf`)” on page 18
 - ◆ “Starting your cluster” on page 18
 - ◆ “Stopping your cluster” on page 19
 - ◆ “Reconfiguring your cluster” on page 19

Two LSF administration commands (`lsfadmin` and `badadmin`)

Only LSF administrators or root can run these commands.

To start and stop LSF, and to reconfigure LSF after making changes to any configuration file, you use two commands:

- ◆ `lsadmin` controls the operation of `lim` and `res`
- ◆ `badadmin` controls the operation of `mbatchd` and `sbatchd` in LSF Batch

Setting up the LSF environment (`cshrc.lsf` and `profile.lsf`)

Before using LSF, you must set up the LSF execution environment.

After logging on to an LSF host, use one of the following shell environment files to set your LSF environment:

- ◆ For example, in `csh` or `tcsh`:


```
% source /usr/share/lsf/lsf_62/conf/cshrc.lsf
```
- ◆ For example, in `sh`, `ksh`, or `bash`:


```
$ . /usr/share/lsf/lsf_62/conf/profile.lsf
```

These two files are created by `lsfinstall` to set up the LSF operating environment.

Starting your cluster

If you have a permanent LSF license

The steps in this section assume you have installed a DEMO LSF license. If you have a permanent LSF license, you must start the license server daemon (`lmgrd`) and verify the status of the license server before using any LSF commands described in this section.

See Chapter 4, “[Working with LSF Licenses](#)” for information about working with a permanent LSF license.

Running LSF as a non-root user

By default, only `root` can start LSF daemons. If `lsfinstall` detected you installed as non-root user, you chose to configure either a multi-user cluster or a single-user cluster:

- ◆ Multi-user—Only `root` can start LSF daemons. Any user can submit jobs to your cluster.

See Chapter 6, “[Troubleshooting LSF Problems](#)” for information about changing ownership and permissions for `lsadmin` and `badadmin`.

To permit LSF administrators to start and stop LSF daemons, set up `/etc/lsf.sudoers`, as described in “[Configuring LSF Startup](#)” on page 38.

- ◆ Single-user—Your user account must be primary LSF administrator. You are able to start LSF daemons, but only your user account can submit jobs to the cluster.

Your user account must be able to read the system kernel information, such as `/dev/kmem`.

lsadmin and badmin

Use `lsadmin` and `badmin` to start the LSF daemons.

- 1 Log on as `root` to each LSF server host.
If you installed a single-user cluster as a non-root user, log on as primary LSF administrator.
Start with the LSF master host, and repeat these steps on all LSF hosts.
- 2 Use the following commands to start the LSF cluster:

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

Before using any LSF commands, wait a few minutes for `lim` on all hosts to do the following:

- ◆ Contact each other
- ◆ Select the master host
- ◆ Exchange initialization information

Stopping your cluster

lsadmin and badmin

Use `lsadmin` and `badmin` to stop the LSF daemons.

- 1 Log on as `root` to any LSF server host.
If you installed a single-user cluster as a non-root user, log on as primary LSF administrator.
- 2 Use the following commands to stop the LSF cluster:

```
# badmin hshutdown all  
# lsadmin resshutdown all  
# lsadmin limshutdown all
```

Reconfiguring your cluster

lsadmin and badmin

Use `lsadmin` and `badmin` to reconfigure LSF after making changes to any configuration file.

- 1 Log on as `root` to any LSF server host.
If you installed a single-user cluster as a non-root user, log on as primary LSF administrator.
- 2 Use the following commands to reconfigure the LSF cluster:
 - ❖ To reload modified LSF configuration files and restart `lim`:

```
# lsadmin reconfig
```
 - ❖ To reload modified LSF Batch configuration files:

```
# badmin reconfig
```
 - ❖ To reload modified LSF Batch configuration files and restart `mbatchd`:

```
# badmin mbdrestart
```

This command also reads the `LSF_LOGDIR/lsb.events` file, so it can take some time to complete if a lot of jobs are running.

See *Administering Platform LSF* for information about which command to run after modifying LSF configuration files.

Checking LSF Status

- Contents
- ◆ “Example command output” on page 21
 - ◆ “Checking cluster configuration (lsadmin)” on page 21
 - ◆ “Finding out cluster status (lsid and lload)” on page 22
 - ◆ “Checking LSF Batch configuration (badmin)” on page 23
 - ◆ “Finding out LSF Batch system status (bhosts and bqueues)” on page 23

Example command output

The LSF commands shown in this section show examples of typical output. The output you see will differ according to your configuration.

The commands are described briefly so that you can easily use them as a “sanity check” for your LSF installation. See the *Platform 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 Chapter 6, “[Troubleshooting LSF Problems](#)” or the *Platform LSF Reference* for help.

Checking cluster configuration (lsadmin)

lsadmin ckconfig -v

The `lsadmin` command controls the operation of an LSF cluster and administers the LSF daemons, `lim` and `res`. Use the `lsadmin ckconfig` command to check the LSF configuration files. The `-v` flag displays detailed information about the LSF configuration:

```
% lsadmin ckconfig -v
Platform LSF 6.2, Oct 18 2005
Copyright 1992-2005 Platform Computing Corporation
Reading configuration from /usr/share/lsf/lsf_62/conf/lsf.conf
Dec 20 12:22:55 2005 24854 5 6.2 /usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/lim -
C

Checking configuration files ...
Dec 20 12:22:55 2005 24854 7 6.2 setMyClusterName: searching cluster files ...
Dec 20 12:22:55 2005 24854 7 6.2 setMyClusterName: local host hosta belongs to
cluster lsf_62
Dec 20 12:22:55 2005 24854 3 6.2 domanager():
/usr/share/lsf/lsf_62/conf/lsf.cluster.lsf_62(13): The cluster manager is the
invoker <lsfadmin> in debug mode
Dec 20 12:22:55 2005 24854 7 6.2 initReadLoad: I have 1 CPUs
Dec 20 12:22:55 2005 24854 6 6.2 Checking Done.
-----
No errors found.
```

The messages shown above are typical output from `lsadmin ckconfig -v`. Other messages may indicate problems with your LSF configuration.

See Chapter 6, “[Troubleshooting LSF Problems](#)” or the *Platform LSF Reference* for help with some common configuration errors.

Finding out cluster status (lsid and lsload)

lsid Tells you if your LSF environment is set up properly. `lsid` displays the current LSF version number, cluster name, and host name of the current LSF master host for your cluster.

The 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
```

```
Platform LSF 6.2, Oct 18 2005
```

```
Copyright 1992-2005 Platform Computing Corporation
```

```
My cluster name is lsf_62
```

```
My master name is hosta
```

If you see the message

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

the `LSF_ENVDIR` environment variable is probably not set correctly. Use `cschrc.lsf` or `profile.lsf` to set up your environment. See “[Top 10 LSF problems](#)” on page 55 for additional help.

lsload Displays the current load levels of the cluster. The output contains one line for each host in the cluster. The status should be `ok` for all hosts in your cluster. For example:

```
% lsload
```

HOST_NAME	status	r15s	r1m	r15m	ut	pg	ls	it	tmp	swp	mem
hostn	ok	0.0	0.0	0.1	1%	0.0	1	224	43M	67M	3M
hostk	-ok	0.0	0.0	0.0	3%	0.0	3	0	38M	40M	7M
hostg	busy	*6.2	6.9	9.5	85%	1.1	30	0	5M	400M	385M
hostf	busy	0.1	0.1	0.3	7%	*17	6	0	9M	23M	28M
hostv	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

```
lsid: ls_getclustername() failed: LIM is down; try later
```

or

```
LSF daemon (LIM) not responding ... still trying
```

after starting or reconfiguring LSF, wait a few seconds and try `lsload` again to give `lim` on all hosts time to initialize. If the problem persists, see “[Top 10 LSF problems](#)” on page 55 for help.

`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. See “[Top 10 LSF problems](#)” on page 55 for more help with this error. See “[About LSF Licenses](#)” on page 40 for more information about LSF licenses.

- 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.

Checking 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 ...

Dec 20 12:22:55 2005 20246 6 6.2 minit: Trying to call LIM to get cluster name
...
Dec 20 12:22:55 2005 20246 6 6.2 Batch is enabled
Dec 20 12:22:55 2005 4433 6 6.2 Checking Done
-----
No errors found.
```

The messages shown above are typical output from `badmin ckconfig -v`. Other messages may indicate problems with the LSF Batch configuration.

See Chapter 6, “[Troubleshooting LSF Problems](#)” or the *Platform LSF Reference* for help with some common configuration errors.

Finding out LSF Batch system status (bhosts and bqueues)

To use LSF Batch commands, the cluster must be up and running. See “[Starting your cluster](#)” on page 18 for information about starting LSF daemons.

bhosts The `bhosts` command tells you if LSF Batch is running properly. `bhosts` displays the status of LSF Batch server hosts in the cluster, and other details about the batch hosts:

- ◆ 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 should be ok for all 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

```
batch system daemon not responding ... still trying
```

after starting or reconfiguring LSF, wait a few seconds and try `bhosts` again to give `mbatchd` time to initialize. If the problem persists, see “[Top 10 LSF problems](#)” on page 55 for help.

bqueues LSF Batch queues organize jobs with different priorities and different scheduling policies.

The `bqueues` command displays the status of available queues and their configuration parameters. For a queue to accept and dispatch jobs, the status should 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
idle	20	Open:Active	-	-	-	-	0	0	0	0

The queue information displayed by `bqueues` is configured in `LSB_CONFDIR/cluster_name/configdir/lsb.queues`. Several queues are defined by default in `lsb.queues`. Modify this file to add, delete, or change queues. See “[Managing Users, Hosts, and Queues](#)” on page 32 for more information about modifying queues.

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

% bqueues -l normal

QUEUE: normal

-- For normal low-priority jobs, 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	-	-	-	-	8	8	0	0	0	0

STACKLIMIT MEMLIMIT

2048 K 5000 K

SCHEDULING PARAMETERS

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

USERS: all users

HOSTS: all hosts used by the LSF Batch system

`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

- ◆ Hosts and users able to use the queue
 - ◆ Scheduling threshold values:
 - ❖ `loadSched` is the threshold for LSF to stop dispatching jobs automatically
 - ❖ `loadStop` is the threshold for LSF to suspend a job automatically
- Other useful commands**
- ◆ The `bparams` command displays information about the LSF Batch configuration parameters.
 - ◆ The `bhist` command displays historical information about jobs.

For more information

- ◆ See *Administering Platform LSF* for more information about seeing the status of your cluster.
- ◆ See the *Platform LSF Reference* for detailed information about the commands described in this section.

Running LSF Jobs

- Contents
- ◆ “Commands for running LSF jobs (`lsrun` and `bsub`)” on page 26
 - ◆ “Submitting batch jobs (`bsub`)” on page 26
 - ◆ “Displaying job status (`bjobs`)” on page 27
 - ◆ “Controlling job execution (`bstop`, `bresume`, `bkill`)” on page 27
 - ◆ “Running interactive tasks (`lsrun` and `lsgrun`)” on page 28
 - ◆ “Integrating your applications with LSF” on page 28

Commands for running LSF jobs (`lsrun` and `bsub`)

You use two basic commands to run jobs through LSF:

- ◆ `bsub` submits jobs to LSF Batch. LSF schedules and dispatches jobs to the best available host based on the scheduling policies you configure in your LSF queues.
- ◆ `lsrun` runs an interactive task on the best available host, based on current system load information gathered by `lim`.

For most jobs, all you need to do is add either `lsrun` or `bsub` in front of the job commands you normally use; there is usually no need to modify job executables or execution scripts.

See the *Platform LSF Reference* for complete usage and command options.

Submitting batch jobs (`bsub`)

bsub The `bsub` command submits jobs to LSF queues.

The following command submits a `sleep` job to the default queue (`normal`):

```
% bsub sleep 60
```

```
Job <3616> is submitted to default queue <normal>.
```

When a job is submitted to LSF, it is assigned a unique job ID, in this case 3616.

You can specify a wide range of job options on the `bsub` command. For example, you can specify a queue, and the job command `sleep` is the last option:

```
% bsub -q short sleep 60
```

```
Job <3628> is submitted to queue <short>.
```

What LSF does with job output

By default, when the job is finished, LSF sends email with a job report and any output and error messages to the user account from which the job was submitted. You can optionally save standard output and standard error to files with the `-o` and `-e` options:

```
% bsub -q short -o /home/user1/job/output.%J -e /home/user1/job/errors.%J ls -l  
Job <3640> is submitted to queue <short>.
```

This command appends the standard output and standard error of the job to the files `output.3640` and `errors.3640` in the `jobs` subdirectory of the home directory of `user1`. The `%J` variable in the command line is replaced by the job ID when the files are created, which helps keep track of job output when you run a lot of jobs.

Interactive batch jobs (`bsub -I`)

To submit an interactive job through LSF, use the `-I` option:

```
% bsub -I ls
```

Submits a batch interactive job that displays the output of the `ls` command.

To submit a batch interactive job by using a pseudo-terminal, use the `bsub -Ip` option. To submit a batch interactive job and create a pseudo-terminal with shell mode support, use the `bsub -Is` option.

Displaying job status (bjobs)

The status of each LSF job is updated periodically, and you can use the job ID to monitor and manipulate the job status.

bjobs The `bjobs` command displays the job ID and other job status. For example:

```
% bjobs
JOBID USER      STAT  QUEUE      FROM_HOST  EXEC_HOST  JOB_NAME    SUBMIT_TIME
1266  user1        RUN   normal     hosta      hostb      sleep 60    Jun
5 17:39:58
```

If all hosts are busy, the job is not started immediately and the `STAT` column says `PEND`. The job `sleep 60` should take 60 seconds to run. When the job completes, LSF sends email reporting the job completion.

Controlling job execution (bstop, bresume, bkill)

bstop To suspend a running job, use the `bstop` command and specify the job ID:

```
% bstop 1266
Job <1266> is being stopped
```

If the job was running when it was stopped, the `bjobs` command shows `USUSP` status for job 1266:

```
% bjobs
JOBID USER      STAT  QUEUE      FROM_HOST  EXEC_HOST  JOB_NAME    SUBMIT_TIME
1266  user1        USUSP normal     hosta      hostb      sleep 60    Jun
5 17:39:58
```

Job owners can only suspend their own jobs. LSF administrators can suspend any job.

bresume To resume a suspended job, use the `bresume` command.

```
% bresume 1266
Job <1266> is being resumed
```

If the job resumes immediately, the `bjobs` command shows `RUN` status for job 1266:

```
% bjobs
JOBID USER      STAT  QUEUE      FROM_HOST  EXEC_HOST  JOB_NAME    SUBMIT_TIME
1266  user1        RUN   normal     hosta      hostb      sleep 60    Jun
5 17:39:58
```

Job owners can only resume their own jobs. LSF administrators can resume any job.

bkill To kill a job, use the `bkill` command, which sends a signal to the specified jobs. For example, if the job owner or the LSF administrator runs the following command, job 1266 is killed:

```
% bkill 1266
Job <1266> is being terminated
```

Running interactive tasks (lsrun and lsgrun)

lsrun The `lsrun` command runs a task on either the current local host or remotely on the best available host, provided it can find the necessary resources and the appropriate host type.

For example, the following command runs the UNIX `ls` command. In this case, the command ran through LSF on the local host:

```
% lsrun ls -l /usr/share/lsf/lsf_62/conf/
total 742
-rw-r--r-- 1 root    lsf      11372 Jul 16 16:23 cshrc.lsf
-rw-r--r-- 1 root    lsf        365 Oct 25 10:55 hosts
lrwxrwxrwx 1 root    lsf         49 Jul 16 15:53 license.dat
drwxr-xr-x 3 lsfadmin lsf        512 Jul 16 15:53 lsbatch
-rw-r--r-- 1 lsfadmin lsf      1776 Nov 23 15:13 lsf.conf
-rw-r--r-- 1 lsfadmin lsf     8453 Nov 16 17:46 lsf.shared
-rw-r--r-- 1 lsfadmin lsf        578 Jul 16 15:53 lsf.task
-rw-r--r-- 1 root    lsf     10485 Jul 16 17:08 profile.lsf
```

You can also specify a host where you want to run a command. For example, the following command runs the UNIX `hostname` command on the remote host `hosta`:

```
% lsrun -v -m hosta hostname
<<Execute hostname on remote host hosta>>
hosta
```

lsgrun The `lsgrun` command is similar to `lsrun`, but it runs a task on a group of hosts.

For example, the following command runs the UNIX `hostname` command on three remote hosts:

```
% lsgrun -v -m "hosta hostb hostc" hostname
<<Executing hostname on hosta>>
hosta
<<Executing hostname on hostb>>
hostb
<<Executing hostname on hostc>>
hostc
```

Integrating your applications with LSF

By integrating your applications with LSF, you can assure your users correct and complete job submission and execution without requiring them to learn LSF commands. There are three basic ways to integrate applications with LSF:

- ◆ Wrapper shell scripts
- ◆ Wrapper binary executables
- ◆ Modifying existing application source code and interfaces

Wrapper shell scripts The easiest integration method is to put the `bsub` command into an executable file like a shell script. A *wrapper script* is an executable file for launching your application through LSF. It gives users a simple interface to run their jobs that is easy to deploy and maintain.

For example, if your application is called `abc`, rename `abc` to `abc_real` and create a wrapper script called `abc`, like this:

```
#!/bin/sh
bsub -R "rusage[abc_license=1:duration=1]" abc_real
```

When users run `abc`, they are actually running a script to submit a job `abc_real` to LSF using 1 shared resource named `abc_license`.

For more information about specifying shared resources using the resource requirement (`rusage`) string on the `-R` option of `bsub`, see Chapter 5, “[Using Shared Resources to Manage Software Licenses](#)”.

By adding appropriate options to the script, you can enhance your integration to:

- ◆ Requeue jobs based on license availability
- ◆ Copy input and output files to and from the local directory on the execution host
- ◆ Calculate and estimate resource requirements

Wrapper binary programs

A *wrapper binary* is similar to a wrapper shell script in the form of a compiled binary executable. Compiled wrapper binaries usually run faster and more efficiently than shell scripts, and they also have access to the LSF API (LSLIB and LSBLIB). Binary code is also more secure because users cannot modify it without the source code and appropriate libraries, but it is more time consuming to develop wrapper binary programs than wrapper shell scripts.

Modifying existing application source code and interfaces

LSF is already integrated closely with many commonly used software products. Platform and other software application vendors provide facilities and services for closer integration of LSF and other applications. By modifying existing application user interfaces, you can enable easy job submission, license maximization, parallel execution, and other advanced LSF features. In some cases, you are able to run an LSF job directly from the application user interface.

We can work with you to discuss your specific application integration requirements. Contact your application vendor or Platform for more information.

For more information

- ◆ See *Administering Platform LSF* for more information about running jobs with LSF.
- ◆ See the *Platform LSF Reference* for information about the `lshrun`, `lshgrun`, `bsub`, and `bjobs` commands.
- ◆ See *Using the Platform LSF SDK* for more information about writing programs that use the LSF API.

Where to go next

Learn about administering your Platform LSF cluster, described in Chapter 3, “[Managing Your Cluster](#)”.

Managing Your Cluster

- Contents**
- ◆ “Managing Users, Hosts, and Queues” on page 32
 - ◆ “Configuring LSF Startup” on page 38

Managing Users, Hosts, and Queues

- Contents**
- ◆ “Making your cluster available to users (cshrc.lsf and profile.lsf)” on page 32
 - ◆ “Adding a host to your cluster” on page 32
 - ◆ “Removing a host from your cluster” on page 35
 - ◆ “Adding a queue” on page 36
 - ◆ “Removing a queue” on page 36

Making your cluster available to users (cshrc.lsf and profile.lsf)

To set up the LSF environment for your users, use the following two shell files:

- ◆ LSF_CONFDIR/cshrc.lsf (for csh, tcsh)
- ◆ LSF_CONFDIR/profile.lsf (for sh, ksh, or bash)

Make sure all LSF users include one of these files at the end of their own .cshrc or .profile file, or run one of these two files before using LSF.

For csh or tcsh Add cshrc.lsf to the end of the .cshrc file for all users:

- ◆ Copy the cshrc.lsf file into .cshrc
- OR
- ◆ Add a line similar to the following to the end of .cshrc:
source /usr/share/lsf/lsf_62/conf/cshrc.lsf

For sh, ksh, or bash Add profile.lsf to the end of the .profile file for all users:

- ◆ Copy the profile.lsf file into .profile
- OR
- ◆ Add a line similar to following to the end of .profile:
. /usr/share/lsf/lsf_62/conf/profile.lsf

Adding a host to your cluster

Task summary Adding a host to your cluster has the following major steps:

- 1 Install LSF binaries for the host type.
- 2 Add host information to lsf.cluster.cluster_name.
- 3 Set up the new host.

Before you start Make sure you have the LSF distribution files for the host types you want to add. For example, to add a Sun SPARC system running 64-bit Solaris 7 or Solaris to your cluster, get the file lsf60_sparc-sol7-64.tar.Z.

Distribution tar files for all supported LSF releases are available for download through the Platform Online Support Web site at www.platform.com or the Platform FTP site <ftp.platform.com>.

Access to the download area of the Platform Web site and the Platform FTP site is controlled by login name and password. If you cannot access the distribution files, send email to support@platform.com.

See the Platform Web site for a complete list of [supported platforms](#).

1 Install LSF binaries for a new host type

Use `lsfinstall` to add new host types to your cluster. You can skip these steps if you already have the executables

- 1 Log on as root to any host that can access the LSF install script directory.
- 2 Change to the LSF install script directory.
For example:

```
# cd /usr/share/lsf/lsf_62/6.2/install
```
- 3 Edit `install.config` to specify desired options for new host types.
You do not need to specify `LSF_LICENSE`.
See the *Platform LSF Reference* for more information about `install.config`.
See *Installing Platform LSF on UNIX and Linux* for information about `lsfinstall`.
- 4 Run `./lsfinstall -f install.config`.
- 5 Follow the steps for host setup in “Completing Your LSF Installation” (`lsf_getting_started.html` file generated by `lsfinstall`) to set up the new hosts.

2 Add host information to `lsf.cluster.cluster_name`

- 1 Log on to the LSF master host as the primary LSF administrator.
- 2 Edit `LSF_CONFDIR/lsf.cluster.cluster_name`, and add host information for the new host to the `Host` section:
 - ❖ Add the name of the host.
 - ❖ Add model or type.
If you enter the `!` keyword in the `model` and `type` columns, the host model is automatically detected by `lim` running on the host.
You may want to use the default values for that host type now, and change them later on when you have more experience or more information. This can be done without interrupting LSF service.
 - ❖ Specify LSF server or client in the `server` column:
 - ❖ 1 (one) indicates an LSF server host
 - ❖ 0 (zero) indicates an LSF client-only host
 By default, all hosts are considered LSF server hosts.

For example:

HOSTNAME	model	type	server	r1m	mem	RESOURCES	REXPRI
hosta	!	SUNSOL6	1	1.0	4	()	0
hostb	!	SUNSOL6	0	1.0	4	()	0
hostc	!	HPPA1132	1	1.0	4	()	0
hostd	!	HPPA1164	1	1.0	4	()	0

End Host

- 3 Save the changes to `LSF_CONFDIR/lsf.cluster.cluster_name`.
- 4 Reconfigure `lim` to enable the new host in the cluster:

```
% lsadmin reconfig
Checking configuration files ...
No errors found.
```

```
Do you really want to restart LIMs on all hosts? [y/n] y
Restart LIM on <hosta> ..... done
Restart LIM on <hostc> ..... done
Restart LIM on <hostd> ..... done
```

The `lsadmin reconfig` command checks for configuration errors.

If no errors are found, you are asked to confirm that you want to restart `lim` on all hosts and `lim` is reconfigured. If fatal errors are found, reconfiguration is aborted.

- 5 Reconfigure `mbatchd`:


```
% badmin reconfig
Checking configuration files ...
No errors found.
Do you want to reconfigure? [y/n] y
Reconfiguration initiated
```

The `badmin reconfig` command checks for configuration errors.

If no fatal errors are found, you are asked to confirm reconfiguration. If fatal errors are found, reconfiguration is aborted.

3 Set up the new host

- 1 Log on as root to any host that can access the LSF install script directory.
- 2 Change to the LSF install script directory.

For example:

```
# cd /usr/share/lsf/lsf_62/6.2/install
```

- 3 Run the `hostsetup` command to set up the new host.

For example:

```
# ./hostsetup --top="/usr/share/lsf/lsf_62" --boot="y"
```

See the *Platform LSF Reference* and *Installing Platform LSF on UNIX and Linux* for information about `hostsetup`.

- 4 Start LSF on the new host:


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

- 5 Run `bhosts` and `lshosts` to verify your changes.

If any host type or host model is UNKNOWN or DEFAULT, see *Administering Platform LSF* to fix the problem.

Adding hosts dynamically

Dynamic host configuration allows you to add hosts to the cluster without manually changing the LSF configuration. See *Administering Platform LSF* for details about adding hosts dynamically.

If you get errors

See Chapter 6, “[Troubleshooting LSF Problems](#)” or the *Platform LSF Reference* for help with some common configuration errors.

Removing a host from your cluster

CAUTION **Never remove the master host from LSF. If you want to remove your current default master from LSF, change `lsf.cluster.cluster_name` to assign a different default master host. Then remove the host that was once the master host.**

- 1 Log on to the host as root or the primary LSF administrator.
- 2 Run `badmin hclose` to close the host. This prevents jobs from being dispatched to the host and allows running jobs to finish.
- 3 When all dispatched jobs are finished, run `lsfshutdown` to stop the LSF daemons on the host to be removed.
- 4 Edit `LSF_CONFDIR/lsf.cluster.cluster_name` and remove the host from the Host section.
- 5 Remove any other references to the host, if applicable, from the following LSF configuration files:
 - ❖ `LSF_CONFDIR/lsf.shared`
 - ❖ `LSB_CONFDIR/cluster_name/configdir/lsb.hosts`
 - ❖ `LSB_CONFDIR/cluster_name/configdir/lsb.queues`
- 6 Log off the host to be removed, and log on as root or the primary LSF administrator to any other host in the cluster.
- 7 Reconfigure lim:

```
% lsadmin reconfig
Checking configuration files ...
No errors found.

Do you really want to restart LIMs on all hosts? [y/n] y
Restart LIM on <hosta> ..... done
Restart LIM on <hostc> ..... done
```

The `lsadmin reconfig` command checks for configuration errors.

If no errors are found, you are asked to confirm that you want to restart `lim` on all hosts and `lim` is reconfigured. If fatal errors are found, reconfiguration is aborted.
- 8 Reconfigure `mbatchd`:

```
% badmin reconfig
Checking configuration files ...
No errors found.
Do you want to reconfigure? [y/n] y
Reconfiguration initiated
```

The `badmin reconfig` command checks for configuration errors.

If no fatal errors are found, you are asked to confirm reconfiguration. If fatal errors are found, reconfiguration is aborted.
- 9 If you configured LSF daemons to start automatically as system startup, remove the LSF section from the host's system startup files. See [“Setting up automatic LSF startup”](#) on page 38 for details.
- 10 If any users of the host use `lscsh` as their login shell, change their login shell to `tcsh` or `csh`. Remove `lscsh` from the `/etc/shells` file.

Removing hosts dynamically Dynamic host configuration allows you to remove hosts from the cluster without manually changing the LSF configuration. See *Administering Platform LSF* for details about removing hosts dynamically.

If you get errors See Chapter 6, “[Troubleshooting LSF Problems](#)” or the *Platform LSF Reference* for help with some common configuration errors.

Adding a queue

Adding a queue does not affect pending or running jobs.

- 1 Log on as the LSF administrator to any host in the cluster.
- 2 Edit `LSB_CONFDIR/cluster_name/configdir/lsb.queues` to add the new queue definition.
You can copy another queue definition from this file as a starting point; remember to change the `QUEUE_NAME` of the copied queue.

See the *Platform LSF Reference* for information about queue configuration in the `lsb.queues` file.

- 3 Save the changes to `lsb.queues`.
- 4 Use the command `badmin ckconfig` to check the new queue definition.
If any errors are reported, fix the problem and check the configuration again.

- 5 When the configuration files are ready, reconfigure the cluster:

```
% badmin reconfig
Checking configuration files ...
No errors found.
Do you want to reconfigure? [y/n] y
Reconfiguration initiated
```

The `badmin reconfig` command checks for configuration errors.

If no fatal errors are found, you are asked to confirm reconfiguration. If fatal errors are found, reconfiguration is aborted.

If you get errors See Chapter 6, “[Troubleshooting LSF Problems](#)” or the *Platform LSF Reference* for help with some common configuration errors.

Removing a queue

View list of existing queues Before removing a queue, make sure there are no jobs in that queue.
Use the `bqueues` command to view a list of existing queues.

If there are jobs in the queue, you must move pending and running jobs to another queue, then remove the queue.

If you remove a queue that has pending jobs in it, the jobs are temporarily moved to a `lost_and_found` queue. The job state does not change. Running jobs continue, and jobs pending in the original queue are pending in the `lost_and_found` queue, and remain pending until the user or the LSF administrator uses the `bswitch` command to switch the jobs into regular queues. Jobs in other queues are not affected.

- Steps**
- 1 Log on to any host in the cluster as primary LSF administrator.
 - 2 Close the queue to prevent any new jobs from being submitted:
For example:

```
% badmin qclose night  
Queue <night> is closed
```

- 3 Move all pending and running jobs into another queue.

For example, the `bswitch -q night` command below chooses jobs from the `night` queue, and the job ID number 0 specifies that all jobs should be switched (job ID of 0 matches all jobs):

```
% bjobs -u all -q night
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
5308	user5	RUN	night	hosta	hostd	job5	Nov 21 18:16
5310	user5	PEND	night	hosta	hostc	job10	Nov 21 18:17

```
% bswitch -q night idle 0
```

```
Job <5308> is switched to queue <idle>
```

```
Job <5310> is switched to queue <idle>
```

- 4 Edit `LSB_CONFDIR/cluster_name/configdir/lsb.queues` and remove or comment out the definition for the queue being removed.
- 5 Save the changes to `lsb.queues`.
- 6 Reconfigure the cluster:

```
% badmin reconfig
```

```
Checking configuration files ...
```

```
No errors found.
```

```
Do you want to reconfigure? [y/n] y
```

```
Reconfiguration initiated
```

The `badmin reconfig` command checks for configuration errors.

If no fatal errors are found, you are asked to confirm reconfiguration. If fatal errors are found, reconfiguration is aborted.

If you get errors See Chapter 6, “[Troubleshooting LSF Problems](#)” or the *Platform LSF Reference* for help with some common configuration errors.

Configuring LSF Startup

- Contents
- ◆ “Allowing LSF administrators to start LSF daemons (`lsf.sudoers`)” on page 38
 - ◆ “Setting up automatic LSF startup” on page 38

Allowing LSF administrators to start LSF daemons (`lsf.sudoers`)

To allow LSF administrators to start and stop LSF daemons, you should configure the `/etc/lsf.sudoers` file. If `lsf.sudoers` does not exist, only `root` can start and stop LSF daemons.

- 1 Log on as `root` to each LSF server host.
Start with the LSF master host, and repeat these steps on all LSF hosts.
- 2 Create `/etc/lsf.sudoers` on each LSF host and specify the `LSF_STARTUP_USERS` and `LSF_STARTUP_PATH` parameters. For example:

```
LSF_STARTUP_USERS="lsfadmin user1"
LSF_STARTUP_PATH=/usr/share/lsf/lsf_62/6.2/sparc-sol2/etc
```

`LSF_STARTUP_PATH` is normally `LSF_SERVERDIR`, where the LSF server binaries (`lim`, `res`, `sbatchd`, `mbatchd`, `mbschd`, etc.) are installed, as defined in your `LSF_CONFDIR/lsf.conf` file.
`lsf.sudoers` must have file permission mode `-rw-----` (600) and be readable and writable only by `root`:

```
# ls -la /etc/lsf.sudoers
-rw-----  1 root    lsf          95 Nov 22 13:57
lsf.sudoers
```
- 3 Run `lsfrestart` to restart the cluster:

```
# lsfrestart
```

Setting up automatic LSF startup

Use the `boot=y` option of `hostsetup` to configure LSF daemons to start automatically on every LSF server host in the cluster.

For more information

- ◆ See *Administering Platform LSF* and the *Platform LSF Reference* for more information about `lsf.sudoers`.
- ◆ See *Administering Platform LSF* for more information about configuring automatic startup.
- ◆ See the *Platform LSF Reference* and *Installing Platform LSF on UNIX and Linux* for information about `hostsetup`.

Where to go next

Learn about Platform LSF licensing, described in Chapter 4, “[Working with LSF Licenses](#)”.

Working with LSF Licenses

- Contents**
- ◆ “About LSF Licenses” on page 40
 - ◆ “Setting up a Permanent LSF License” on page 42

About LSF Licenses

- Contents ♦ “Types of LSF licenses” on page 40
- ♦ “Where the license file is located” on page 41

Types of LSF licenses

LSF uses two types of licenses:

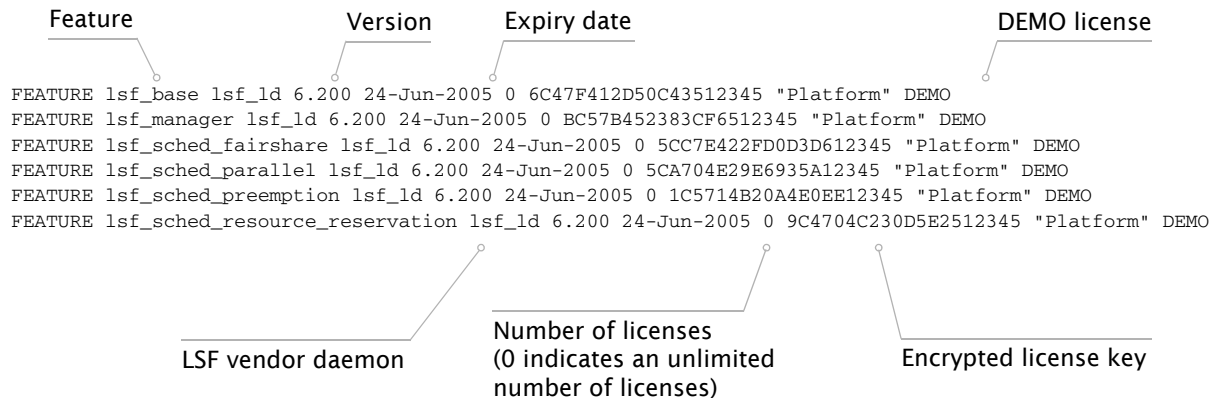
- ♦ File-based DEMO licenses, which do not need a license server.
These are typically used while evaluating LSF and usually expire after 30 days. Each FEATURE line in the license contains an expiry date and ends with DEMO. A DEMO license usually licenses all LSF features and allows an unlimited number of LSF servers and clients.
- ♦ Server-based permanent licenses, which require FLEXlm 7.2 or later and are used in production installations.
Permanent licenses do not expire; they license only the LSF features you buy for a specific number of hosts. Each FEATURE line in the license ends with Platform.

Licenses are stored in a text file. The default name of the license file is `license.dat`. During installation, you specify the location of the license for `lsfinstall` to install.

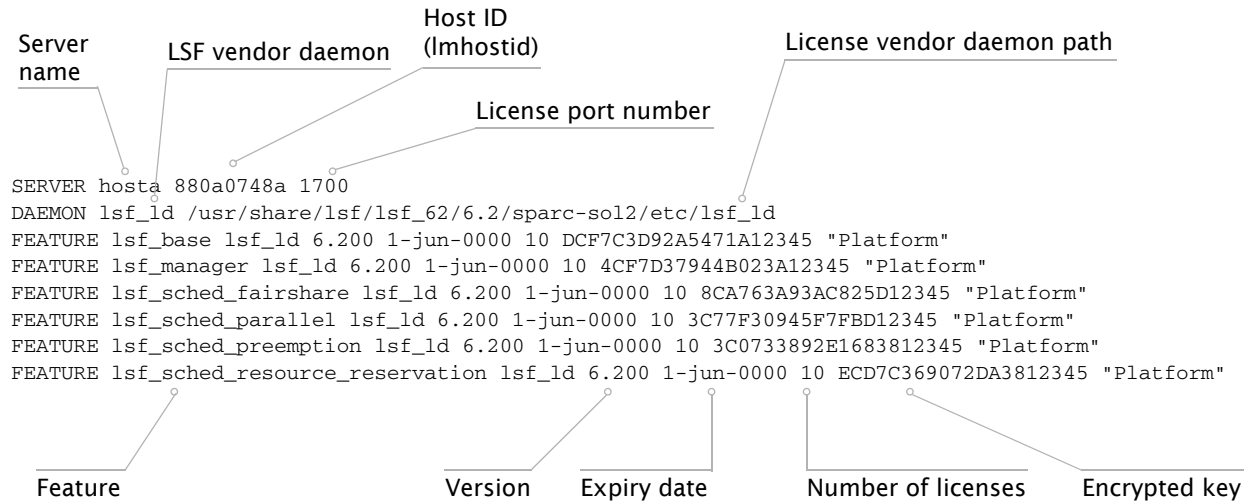
New installations typically use a DEMO license to install LSF and get it running, then switch to a permanent license later, as described in “[Setting up a Permanent LSF License](#)” on page 42.

Example DEMO license file

The following is an example of a DEMO license file. This file contains one FEATURE line for each LSF feature. The license is for Platform LSF Version 6.2 and is valid until June 24, 2005.



Example permanent license file



In this permanent license, the license server is configured to run on `hosta`, using TCP port 1700. It allows 10 hosts to run the following features, with no expiry date:

License FEATURE	PRODUCT in <code>lsf.cluster.cluster_name</code>
<code>lsf_base</code>	LSF_Base
<code>lsf_manager</code>	LSF_Manager
<code>lsf_sched_fairshare</code>	LSF_Sched_Fairshare
<code>lsf_sched_parallel</code>	LSF_Sched_Parallel
<code>lsf_sched_preemption</code>	LSF_Sched_Preemption
<code>lsf_sched_resource_reservation</code>	LSF_Sched_Resource_Reservation

Where the license file is located

The `LSF_LICENSE_FILE` parameter in `LSF_CONFDIR/lsf.conf` points to the license file. You should put the license file in a shared directory to make it accessible from any LSF host that can become the LSF master host.

See the *Platform LSF Reference* for information about the format of the `LSF_LICENSE_FILE` parameter.

Setting up a Permanent LSF License

- Contents
- ◆ “Getting information needed for a permanent license” on page 42
 - ◆ “Getting a license key” on page 42
 - ◆ “Example permanent license file” on page 41
 - ◆ “Preparing and installing a permanent license” on page 43
 - ◆ “Starting the license server daemon (lmgrd)” on page 42
 - ◆ “Checking the license status (lmstat)” on page 44
 - ◆ “Updating an existing permanent license” on page 45
 - ◆ “If you have problems” on page 45

See *Administering Platform LSF* for more information about working with LSF licenses.

Getting information needed for a permanent license

To install LSF software for production use, choose a host as an LSF license server and get a permanent license from Platform or your LSF vendor. You must supply a hardware host name and host identifier for each license server host at your site.

lmhostid Use the `lmhostid` command to get the hardware identifier for your license server. For example:

```
# lmhostid
lmhostid - Copyright (C) 1989-2005 Globetrotter Software, Inc.
The FLEXlm host ID of this machine is "68044d20"
The lmhostid command is in LSF_SERVERDIR. For example:
/usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/lmhostid
```

Getting a license key

Send the host name and host identifier of the license server host to Platform at license@platform.com or your LSF vendor.

Changing your license server host To change your license server to another host, run `lmhostid` on the new license server and send the output of to license@platform.com or your LSF vendor.

Starting the license server daemon (lmgrd)

The license server daemon `lmgrd` must be running before you start LSF daemons. The path to the LSF vendor license daemon `lsf_ld` must be specified in the `DAEMON` line of the license file. For example:

```
DAEMON lsf_ld /usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/lsf_ld
```

lmgrd To start the license daemons:

- 1 Log on to the license server host as LSF administrator. Do not run `lmgrd` as root.
- 2 Use the `lmgrd` command to start the license server daemon. For example:


```
% lmgrd -c /usr/share/lsf/lsf_62/conf/license.dat -l /usr/share/lsf/lsf_62/log/license.log
```

The `lmgrd` command is in `LSF_SERVERDIR`. For example:

```
/usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/lmgrd
```

See the `lmgrd(8)` man page for information about FLEXlm commands.

Multiple license servers

If you use multiple FLEXlm license servers, start `lmgrd` on all license servers.

If `lmgrd` does not start

If you get the message

```
port already in use
```

the license port number defined in `LSF_LICENSE_FILE` or in `LSF_CONFDIR/license.dat` is in use by another application. The default port is 1700.

This could mean:

- ◆ `lmgrd` is already running for this license
Use `ps -ef` and make sure that `lmgrd` and `lsf_ld` are not running.
- ◆ `lmgrd` has been stopped and the operating system has not cleared the port
Wait a few minutes for the port to clear.
- ◆ Another process is using the port
This is not likely, but if so, change the port number in `LSF_LICENSE_FILE` or in `LSF_CONFDIR/license.dat`.
See “[Setting the FLEXlm server TCP port in license.dat](#)” on page 44 to change the port number.

Preparing and installing a permanent license

These steps assume the license server is already running. When you receive your permanent license:

- 1 Copy it to a file called `license.dat`.
If you receive your license by post or FAX, use a text editor to create a `license.dat` file.
- 2 Make sure that the `LSF_LICENSE_FILE` parameter in `LSF_CONFDIR/lsf.conf` specifies the full path to the file `license.dat`. For example:

```
LSF_LICENSE_FILE=/usr/share/lsf/lsf_62/conf/license.dat
```


The FLEXlm license server must be able to access this file.
- 3 Edit the `DAEMON` line in the license file to point to the LSF vendor license daemon `lsf_ld`. The LSF vendor license daemon is installed in `LSF_SERVERDIR`.
For example:

```
DAEMON lsf_ld /usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/lsf_ld
```


The FLEXlm license server must be able to access this file.
- 4 Verify that the `FEATURE` lines in the license file (`LSF_CONFDIR/license.dat`) match the features enabled in the `PRODUCTS` line in `LSF_CONFDIR/lsf.cluster.cluster_name`.
For example, if your license has the following `FEATURE` lines:

```
FEATURE lsf_base lsf_ld 6.200 1-jun-0000 10 DCF7C3D92A5471A12345 "Platform"
```

```
FEATURE lsf_manager lsf_ld 6.200 1-jun-0000 10 4CF7D37944B023A12345 "Platform"
```

the PRODUCTS line in LSF_CONFDIR/lsf.cluster.cluster_name must contain:

```
PRODUCTS=LSF_Base LSF_Manager
```

If you do not have licenses for some features in the PRODUCTS line, contact Platform at license@platform.com or your LSF vendor. To continue installing your permanent license, remove the unlicensed features from the PRODUCTS line.

- 5 To make the new permanent license take effect and reconfigure the cluster with the commands:
 - ❖ `lsadmin reconfig`
 - ❖ `badmin reconfig`
- 6 After the cluster starts, use `lsid` and `bhosts` to make sure LSF is up and running.

Setting the FLEXlm server TCP port in license.dat

The default port set by FLEXlm is 1700. If this port is already in use by another application, choose an unused port number for the TCP port number that the FLEXlm server uses. If the `lmgrd` daemon complains that the license server port is in use, you can choose another port number.

For example, if your license file contains the line:

```
SERVER hosta 68044d20 1700
```

and you want your FLEXlm server to use TCP port 3883, change the SERVER line to:

```
SERVER hosta 68044d20 3883
```

After changing the license file, use the `lmreread` command to reread the license file. For example:

```
% lmreread -c /usr/share/lsf/lsf_62/conf/license.dat -l  
/usr/share/lsf/lsf_62/log/license.log
```

The `lmreread` command is in LSF_SERVERDIR. For example:

```
/usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/lmreread
```

Checking the license status (lmstat)

If you are using a permanent LSF license, check the license server by starting the license manager daemon and displaying the status with the `lmstat` command.

The `lmstat` command is in LSF_SERVERDIR. For example:

```
/usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/lmstat
```

If you are using a DEMO license, you do not have to perform this check because a DEMO license does not run a license server daemon.

- lmstat** After starting the license server, use the `lmstat` command to check its status and display the number of licenses available. You must use the `-c` option of `lmstat` to specify the path to the LSF license file.

The output of `lmstat` gives the status of:

- ◆ The license server (`lmgrd`)
- ◆ The LSF vendor daemon (`lsf_ld`)
- ◆ The number of available licenses for each feature in the license file

For example, depending on the LSF features installed, the output of the command should look something like the following:

```
% lmstat -a -c /usr/share/lsf/lsf_62/conf/license.dat
lmstat - Copyright (C) 1989-2000 Globetrotter Software, Inc.
Flexible License Manager status on Fri 3/15/2005 08:39

License server status: 1700@hosta
  License file(s) on hosta: /usr/share/lsf/lsf_62/conf/license.dat:

  hosta: license server UP (MASTER) v7.0

Vendor daemon status (on hosta):

  lsf_ld: UP v7.0

Feature usage info:

Users of lsf_base: (Total of 4 licenses available)

Users of lsf_manager: (Total of 4 licenses available)
```

Updating an existing permanent license

You need Platform to update your permanent license every time you:

- ◆ Increase the number of licenses for your cluster
- ◆ Add a separately licensed feature (e.g., License Maximizer) or enable the feature on additional hosts
- ◆ Upgrade between major releases (e.g., LSF 5.1 to LSF 6.2)
- ◆ Upgrade between maintenance releases (e.g., LSF 6.1 to LSF 6.2)
- ◆ Change your FLEXLM license server host or its host name or TCP port number

Contact Platform at license@platform.com or your LSF vendor and get an INCREMENT license key or a new FEATURE line, and follow the steps in “[Preparing and installing a permanent license](#)” on page 43 to update your license.

You do not need to update your permanent license when you:

- ◆ Remove hosts
- ◆ Replace hosts without increasing the number of licenses required

If you have problems

- ◆ Check the `LSF_LOGDIR/license.log` file for error messages
- ◆ See problem “[2 Host does not have a software license](#)” on page 55 in Chapter 6, “[Troubleshooting LSF Problems](#)” for possible solutions
- ◆ Contact Platform Support or support@platform.com

For more information

- ◆ See *Administering Platform LSF* for more information about working with LSF licenses and license servers.

- ◆ See the *FLEXlm End Users Guide*, available for download from GLOBEtrouter Software, Inc. at www.globetrotter.com, for more information about FLEXlm.

Where to go next

Learn how to set up an LSF External LIM (ELIM) to monitor dynamic shared resources, described in Chapter 5, “[Using Shared Resources to Manage Software Licenses](#)”.

Using Shared Resources to Manage Software Licenses

Contents ♦ “Managing Software Licenses and Other Shared Resources” on page 48

Managing Software Licenses and Other Shared Resources

This chapter uses managing software licenses as an example of how to set up an LSF External LIM (ELIM) to monitor dynamic shared resources.

- Contents**
- ◆ “About dynamic shared resources” on page 48
 - ◆ “Defining dynamic shared resources in an ELIM” on page 48
 - ◆ “Mapping dynamic shared resources to hosts” on page 49
 - ◆ “Monitoring dynamic shared resources” on page 49
 - ◆ “Using dynamic shared resources” on page 50

About dynamic shared resources

LSF keeps track of two main types of resources:

- ◆ Host-based resources are available on all hosts in the cluster; for example, host type and model, or node-locked software licenses.
- ◆ Shared resources are managed as dynamic load indexes available for a group of hosts in the cluster; for example, networked floating software licenses, shared file systems.

Shared resources are shared by a group of LSF hosts. LSF manages shared resources for the purpose of host selection and batch or interactive job execution. These resources are *dynamic* because the load on the system changes with the availability of the resources.

Software licenses as shared resources

The most common application of shared resources is to manage software licenses. Configuring your application licenses as shared resources lets you submit all your jobs at any time, and have LSF run the jobs according to their priorities when licenses are available, or queue them when licenses are not available, then dispatch them when licenses are free. This ensures optimal use of such costly and critical resources.

Defining dynamic shared resources in an ELIM

For LSF to use a shared resource like a software license, you must define the resource in the Resource section of the `lsf.shared` file. You define the type of resource and how often LSF should refresh the value of the resource.

For LSF to track the resources correctly over time, you must define them as external load indices so that LSF can update them periodically with a program called an External Load Information Manager (ELIM).

An ELIM can be a shell script or a compiled binary program, which returns the values of the shared resources you define. The ELIM must be named `elim` and located in `LSF_SERVERDIR`. For example:

```
/usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/elim
```

You can find examples of sample ELIMs in the `misc/examples` directory.

Example: shared licenses

In `lsf.shared`, define two dynamic shared resources for software licenses, named `license1` and `license2`:

Begin Resource

RESOURCENAME	TYPE	INTERVAL	INCREASING	RELEASE	DESCRIPTION	# Keywords
license1	Numeric	30	N	Y	(license1 resource)	
license2	Numeric	30	N	Y	(license2 resource)	

End Resource

- ◆ The TYPE of shared resource can be:
 - ❖ Numeric
 - ❖ Boolean
 - ❖ String
 In this case, the resource is numeric.
- ◆ The INTERVAL specifies how often the value should be refreshed; in this case, the ELIM updates the shared resource values every 30 seconds.
- ◆ The N in INCREASING means that the license resources are decreasing; that is, as more licenses become available, the load becomes lower.
- ◆ The Y in RELEASE means that the license resources are released when a job using the license is suspended.

Mapping dynamic shared resources to hosts

To make LSF aware of where the defined dynamic shared resources `license1` and `license2` you just defined, you map them to the hosts where they are located.

In `LSF_CONFDIR/lsf.cluster.cluster_name`, configure a `ResourceMap` section to specify the mapping between shared resources `license1` and `license2` you defined in `LSF_CONFDIR/lsf.shared`, and the hosts you want to map them to. For example:

```
Begin ResourceMap
RESOURCENAME  LOCATION
license1      [all]
license1      [all]
End ResourceMap
```

In this resource map, `[all]` under `LOCATION` means that resources `license1` and `license2` under `RESOURCENAME` are available on all hosts in the cluster. Only one ELIM needs to run on the master host because the two resources are the same for all hosts. If the location of the resources is different on different hosts, an ELIM must run on every host.

Monitoring dynamic shared resources

For LSF to receive external load indices correctly, the ELIM must send a count of the available resources to standard output in the following format:

```
number_indexes [index_name index_value] ...
```

For example:

```
2 license1 3 license2 2
```

Where the fields contain the following information:

- ◆ The total number of external load indices (2)
- ◆ The name of the first external load index (`license1`)
- ◆ The value of the first load index (3)
- ◆ The name of the second external load index (`license2`)

- ◆ The value of the second load index (2) and so on.

Writing the ELIM program

The ELIM must be an executable program, named `elim`, located in the `LSF_SERVERDIR` directory.

When `lim` is started or restarted, it invokes `elim` on the same host and takes the standard output of the external load indices sent by `elim`. In general, you can define any quantifiable resource as an external load index, write an ELIM to report its value, and use it as an LSF resource.

Example `elim`

The following sample `elim` uses `license1` and `license2` and assumes the FLEXlm license server controls them:

```
#!/bin/sh
NUMLIC=2          # number of dynamic shared resources
while true
do
TMPLICS='/usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/lic -c
/usr/share/lsf/lsf_62/conf/license.dat -f license1, license2'

LICS='echo $TMPLICS | sed -e s/-/_/g'
echo $NUMLIC $LICS # $NUMLIC is number of dynamic shared
resource
sleep 30          # Resource
done
```

The `sed` command changes the minus sign (-) to underscore (_) in the license feature names because LSF uses the minus sign for calculation, and it is not allowed in resource names.

`lic` utility

The `lic` utility is available from support@platform.com. You can also use the FLEXlm command `lmstat` to make your own ELIM.

Using dynamic shared resources

To enable the new shared resources in your cluster, restart LSF with the commands:

- ◆ `lsadmin reconfig`
- ◆ `badmin reconfig`

If no errors are found, use `lsload -l` to verify the value of your dynamic shared resources:

HOST_NAME	status	r15s	r1m	r15m	ut	pg	io	ls	it	tmp	swp	mem	license1	license2
hosta	ok	0.1	0.3	0.4	8%	0.2	50	73	0	62M	700M	425M	3	0
hostb	ok	0.1	0.1	0.4	4%	5.7	3	3	0	79M	204M	64M	3	0

Submitting jobs using shared resources

To submit a batch job using one `license1` resource, use a command like:

```
% bsub -R 'rusage[license1=1:duration=1]' myjob
```

In the resource requirement (`rusage`) string, `duration=1` means that `license1` is reserved for 1 minute to give LSF time to check it out from FLEXlm.

You can also specify the resource requirement string at queue level, in the `RES_REQ` parameter for the queue. In

`LSB_CONFDIR/cluster_name/configdir/lsb.queue`s, specify:

```
Begin Queue
QUEUE_NAME = license1
RES_REQ=rusage[license1=1:duration=1]
...
End Queue
```

Then submit a batch job using one `license1` resource using a command like:

```
% bsub -q license1 myjob
```

When licenses are available, LSF runs your jobs right away; when all licenses are in use, LSF puts your job in a queue and dispatches them as licenses become available. This way, all of your licenses are used to the best advantage.

For more information

See *Administering Platform LSF* for more information about adding external resources to your cluster and using an ELIM to customize resources.

See the *Platform LSF Reference* for more information about `lsf.shared` and `lsf.cluster.cluster_name` and the parameters for configuring shared resources.

Troubleshooting LSF Problems

This chapter covers solutions to the top 10 LSF problems in the order that you would most likely encounter them as you begin to use LSF. If you cannot find a solution to your problem here, contact your Platform system engineer or support@platform.com.

Contents ♦ “Common LSF Problems” on page 54

Common LSF Problems

- Contents
- ◆ “Finding LSF error logs” on page 54
 - ◆ “For most LSF problems” on page 54
 - ◆ “Top 10 LSF problems” on page 55

Finding LSF error logs

When something goes wrong, LSF server daemons log error messages in the LSF log directory (LSF_LOGDIR).

Make sure that the primary LSF administrator owns LSF_LOGDIR, and that root can write to this directory. If an LSF server is unable to write to LSF_LOGDIR, then the error logs are created in /tmp.

LSF logs to the following files

- ◆ `lim.log.host_name`
- ◆ `res.log.host_name`
- ◆ `pim.log.host_name`
- ◆ `mbatchd.log.master_host`
- ◆ `mbschd.log.master_host`
- ◆ `sbatchd.log.host_name`

If there are any error messages in the log files that you do not understand, contact your Platform system engineer or support@platform.com.

For most LSF problems

The general troubleshooting steps for most LSF problems are:

- 1 Run `lsadmin ckconfig -v` and note any errors shown in the command output. Look for the error in “Top 10 LSF problems” on page 55. If none of these applies to your situation, contact support@platform.com.

- 2 Use the following commands to restart the LSF cluster:

```
# lsadmin limrestart all
# lsadmin resrestart all
# badmin hrestart all
```

- 3 Run `ps -ef` to see if the LSF daemons are running.

Look for the processes similar to the following:

```
root 17426      1  0  13:30:40 ?    0:00 /usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/lim
root 17436      1  0  13:31:11 ?    0:00 /usr/share/lsf/lsf_62/6.2/sparc-
sol2/etc/sbatchd
root 17429      1  0  13:30:56 ?    0:00 /usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/res
```

- 4 Check the LSF error logs on the first few hosts listed in the Host section of `LSF_CONFDIR/lsf.cluster.cluster_name`. If `LSF_MASTER_LIST` is defined in `LSF_CONFDIR/lsf.conf`, check the error logs on the hosts listed in this parameter instead.

Top 10 LSF problems

1 Cannot open lsf.conf file

You might see this message running `lsid`. This means that `LSF_CONFDIR/lsf.conf` is not accessible to LSF.

By default, LSF checks the directory defined by `LSF_ENVDIR` for `lsf.conf`. If `lsf.conf` is not in `LSF_ENVDIR`, LSF looks for it in `/etc`.

- ◆ Make sure that there is either a symbolic link from `/etc/lsf.conf` to `lsf.conf` or
- ◆ Use `csrhc.lsf` or `profile.lsf` to set up your LSF environment. Make sure that `csrhc.lsf` or `profile.lsf` are available for users to set the LSF environment variables.

See “[Setting up the LSF environment \(csrhc.lsf and profile.lsf\)](#)” on page 18 for more information.

2 Host does not have a software license

You might see this message when you run `lsid`, `lsload`, and other `ls*` commands. It means that the `LSF_CONFDIR/license.dat` file has errors or the path to the license file is incorrect.

Some typical license problems:

- ◆ The license file was edited on Windows and incorrect line ending characters (^M) are present in the file. Edit `LSF_CONFDIR/license.dat` on UNIX and remove the ^M characters.
- ◆ FEATURE lines in the license file have line breaks:
 - ❖ FEATURE lines in a DEMO LSF license must be on one line, with no line breaks, begin with `FEATURE` and end with `DEMO`
 - ❖ FEATURE lines in a permanent LSF license must be on one line, with no line breaks, begin with `FEATURE` and end with `Platform`.

Edit `LSF_CONFDIR/license.dat` and remove the line breaks. For example:

```
FEATURE lsf_base lsf_ld 6.200 1-jun-0000 100 DCF7C3D92A5471A12345 "Platform"
```

- ◆ Your DEMO license has expired. Get a new DEMO license, or install a permanent license, as described in “[Setting up a Permanent LSF License](#)” on page 42.
- ◆ You have a permanent license and the license server daemon (`lmgrd`) is not running. Start the license server as described in “[Starting the license server daemon \(lmgrd\)](#)” on page 42.
- ◆ You have installed a new license but an unlicensed `lim` is still running on the LSF master host.

Run `ps -ef` to see if `lim` is running.

Restart `lim` as described in “[Starting your cluster](#)” on page 18.

- ◆ The FEATURE lines in the license file (`LSF_CONFDIR/license.dat`) do not match the features enabled in the `PRODUCTS` line in `LSF_CONFDIR/lsf.cluster.cluster_name`.

For example, if your license has the following FEATURE lines:

```
FEATURE lsf_base lsf_ld 6.200 1-jun-0000 10 DCF7C3D92A5471A12345 "Platform"
FEATURE lsf_manager lsf_ld 6.200 1-jun-0000 10 4CF7D37944B023A12345 "Platform"
```

the PRODUCTS line in `LSF_CONFDIR/lsf.cluster.cluster_name` must contain:

```
PRODUCTS=LSF_Base LSF_Manager
```

Modify the PRODUCTS line to fix the error.

See Chapter 4, “[Working with LSF Licenses](#)” for information about working with a permanent LSF license.

- ◆ `lsf.conf` is not in the location specified by `LSF_ENVDIR`.
Check that `LSF_LICENSE_FILE` parameter in `lsf.conf` is correct. `LSF_LICENSE_FILE` can be either:
 - ❖ The full path name to the license file
For example:

```
LSF_LICENSE_FILE=/usr/share/lsf/lsf_62/conf/license.dat
```
 - ❖ For a permanent license, the name of the license server host and the port number used by `lmgrd` in the form `port_number@host_name`.
For example, if your license file contains the line:

```
SERVER hosta 68044d20 1700
```


`LSF_LICENSE_FILE` would be:

```
LSF_LICENSE_FILE=1700@hosta
```

Permanent license problems:

- ◆ Run `lmhostid` on your license server host, and check that the host ID matches the first line of your license file.

The `lmhostid` command is in `LSF_SERVERDIR`. For example:

```
/usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/lmhostid
```

- ◆ Check the path to the LSF vendor daemon `lsf_ld` in the license file. The path must be specified in the DAEMON line of the license file. For example:

```
DAEMON lsf_ld /usr/share/lsf/lsf_62/6.2/sparc-sol2/etc/lsf_ld
```

- ◆ If there are multiple license management daemons running on your machine, or the default license port 1700 is being used by another process, edit the license file and change the port number as described in “[Setting the FLEXlm server TCP port in license.dat](#)” on page 44.
- ◆ Use `lmstat` to check if you have enough licenses for your cluster. Each CPU on LSF server hosts (not each server host itself) in the cluster uses one LSF server license. Each LSF client host (not each CPU) uses one LSF client license.
- ◆ After making any changes to your permanent license, stop `lmgrd` first, then stop `lsf_ld` before restarting `lmgrd`; otherwise, `lmgrd` will start another `lsf_ld` right away and `lsf_ld` will not restart correctly.

3 LIM unavailable Sometimes `lim` is up, but the `lsload` command displays one of the following error messages:

- ◆ `lsid: ls_getclustername() failed: LIM is down; try later`
- ◆ `LSF daemon (LIM) not responding ... still trying`
- ◆ `ls_limcontrol: Communication time out`

If `lim` has just been started, this is normal; `lim` needs time to read configuration files and contact `lim` daemons on other hosts. If `lim` does not respond within one or two minutes, check the `lim` error log (`LSF_LOGDIR/lim.log.host_name`) for the host you are working on.

When the local `lim` is running but there is no master `lim` in the cluster, LSF applications display the following message:

```
Cannot locate master LIM now, try later.
```

`lim` problems can have several causes. Check:

- ◆ The host is defined in the configuration files:
 - The host must be listed in the `HOSTNAME` column of the Host section in `LSF_CONFDIR/lsf.cluster.cluster_name`
 - ❖ If the host is an LSF server:
 - The value in the `SERVER` column of the Host section in `LSF_CONFDIR/lsf.cluster.cluster_name` must be 1.
 - ❖ If the host is an LSF client:
 - ❖ No LSF daemons should be running on the host
 - ❖ The value in the `SERVER` column of the Host section in `LSF_CONFDIR/lsf.cluster.cluster_name` must be 0.
 - ❖ `LSF_SERVER_HOSTS` variable is set in `LSF_CONFDIR/lsf.conf`
- ◆ The LSF environment (`LSF_ENVDIR`) is set properly.
 - Use `cschrc.lsf` or `profile.lsf` to set the LSF environment on the host.
- ◆ LSF licensing.
 - Run `lsadmin ckconfig -v` and correct the problems shown in the command output.
 - See problem “[2 Host does not have a software license](#)” on page 55 and Chapter 4, “[Working with LSF Licenses](#)” for more information.
- ◆ Ownership of the LSF files and directories.
 - The LSF primary administrator should own all LSF directories and most files. The LSF administration commands `lsadmin` and `badmin` must be owned by `root` and have the file permission mode `-rwsr-xr-x` (user ID bit for the owner is `setuid`).
 - See problem “[5 lsadmin or badmin fails](#)” on page 59 for more information.
- ◆ The ports for `lim` and `res` are not used by any other process, and are set correctly in one of:
 - ❖ `LSF_CONFDIR/lsf.conf`
 - ❖ `/etc/services`
 - ❖ NIS

A common mistake is a typo in port names in `LSF_CONFDIR/lsf.conf`.
By default, the following ports are defined in `LSF_CONFDIR/lsf.conf`:

 - ❖ `LSF_LIM_PORT=6879`
 - ❖ `LSF_RES_PORT=6878`
 - ❖ `LSB_MBD_PORT=6881`

❖ `LSB_SBD_PORT=6882`

The port numbers can be any numbers ranging from 1024 to 65535 that are not already used by other services. To make sure that the port numbers you supply are not already used by applications registered in your service database, check `/etc/services`.

To change the port numbers:

- a Shut down your cluster.
- b Edit `LSF_CONFDIR/lsf.conf`.
- c Restart LSF.

4 LSF daemons not responding

After starting the LSF daemons, use the `ps` command to check that LSF is running on all LSF hosts, and `mbatchd` is running on the master host. `mbatchd` is started automatically by `sbatchd` on the master host. You never need to start `mbatchd` yourself.

Before using any LSF commands, wait a few minutes `lim` on all hosts to do the following:

- ◆ Contact each other
- ◆ Select a master host
- ◆ Exchange initialization information

Check the daemon error logs. Most common problems are:

- ◆ LSF service port conflicts:

Check that the ports for `lim` and `res` are not used by any other process, and are set correctly in one of:

- ❖ `LSF_CONFDIR/lsf.conf`
- ❖ `/etc/services`
- ❖ NIS

and are not used by any other process.

A common mistake is a typo in port names in `LSF_CONFDIR/lsf.conf`.

By default, the following ports are defined in `LSF_CONFDIR/lsf.conf`:

- ❖ `LSF_LIM_PORT=6879`
- ❖ `LSF_RES_PORT=6878`
- ❖ `LSB_MBD_PORT=6881`
- ❖ `LSB_SBD_PORT=6882`

The port numbers can be any numbers ranging from 1024 to 65535 that are not already used by other services. To make sure that the port numbers you supply are not already used by applications registered in your service database, check `/etc/services`.

To change the port numbers:

- a Shut down your cluster.
 - b Edit `LSF_CONFDIR/lsf.conf`.
 - c Start LSF.
- ◆ LSF license problem.

Run `lsadmin ckconfig -v` and correct the problems shown in the command output.

See problem “[2 Host does not have a software license](#)” on page 55 and Chapter 4, “[Working with LSF Licenses](#)” for more information.

- ◆ Ownership of the LSF files and directories.

The LSF primary administrator should own all LSF directories and most files.

In particular, `LSB_SHARED_DIR` (e.g., `/usr/share/lsf/lsf_62/work`) must be owned and writable by the LSF primary administrator.

The LSF administration commands `lsadmin` and `badmin` must be owned by `root` and have the file permission mode `-rwsr-xr-x` (user ID bit for the owner is `setuid`).

See problem “[5 lsadmin or badmin fails](#)” on page 59 for more information.

5 lsadmin or badmin fails

The LSF administration commands `lsadmin` and `badmin`, or the `eauth` executable might give the error messages like:

- ◆ User permission denied.
- ◆ Operation not permitted.

Check the following:

- ◆ If you ran `lsfinstall` as `root`, `lsfadmin`, `badmin`, and `eauth` have the file permission mode `-rwsr-xr-x` (4755) so that the user ID bit for the owner is `setuid`. However, on the file system where LSF was installed, `setuid` permission may be turned off. Do one of the following:
 - ❖ If `lsadmin`, `badmin`, and `eauth` are in a directory shared through NFS, share and mount the `LSF_TOP` directory with `setuid` enabled. *Do not* mount with the `nosuid` flag.
 - ❖ If your site does not permit this, you must copy `lsadmin` and `badmin`, and `eauth` to a local directory on each host in the cluster. (make sure the local directory is in the `PATH`; for example, `/usr/bin` or `/bin`.)
- ◆ If you ran `lsfinstall` as a non-root user to install a multi-user cluster, you must manually change the ownership for `lsadmin` and `badmin` to `root` and the file permission mode to `-rwsr-xr-x` (4755) so that the user ID bit for the owner is `setuid`.

Use the following commands to set the correct owner, user ID bit, and file permission mode for a multi-user cluster:

```
# chown root lsadmin badmin eauth
# chmod 4755 lsadmin badmin eauth
```

- ◆ You may also see these messages if you run the `bsub` command as `root`.
By default, you cannot submit jobs with `bsub` as `root`. This is a normal security precaution. If you need to submit jobs as `root`, set `LSF_ROOT_REX=local` in `LSF_CONFDIR/lsf.conf`.

If you continue to see this message, it means that the remote host cannot securely determine the user ID of the user requesting remote execution.

Check:

- ❖ The messages in `LSF_LOGDIR/res.log.host_name` on the execution host. `res` is responsible for authenticating users in LSF
- ❖ The setting of LSF authentication (`LSF_AUTH` in `LSF_CONFDIR/lsf.conf`):
 - ❖ LSF default authentication is `eauth` (`LSF_AUTH` is not or is defined as `eauth` in `lsf.conf`)
 - ❖ If `LSF_AUTH` is defined as `identd` in `lsf.conf`, `identd` must be installed and running properly on each LSF host
- ❖ That the `eauth` executable in `LSF_SERVERDIR` is implemented correctly
- ❖ That all LSF commands are owned by `root` and have the file permission mode to `-rwsr-xr-x` so that the user ID bit for the owner is `setuid`

See *Administering Platform LSF* for information about LSF authentication and the `eauth` executable.

6 Submitted jobs do not start

Many things can prevent jobs from starting. Check:

- ◆ Wait for up to a minute for LSF to dispatch your job. Some LSF activities happen periodically. You may be able to speed up job dispatch by changing the following parameters:
 - ❖ `JOB_ACCEPT_INTERVAL` in `LSB_CONFDIR/cluster_name/configdir/lsb.queues`
 - ❖ `SBD_SLEEP_TIME` in `LSB_CONFDIR/cluster_name/configdir/lsb.params`

See the *Platform LSF Reference* for information about these parameters.

- ◆ Use `bjobs -p` and check the job pending reason.
- ◆ Examine the output of `bhosts -l` for the load thresholds on the hosts and `lsload -E` for the effective run queue lengths on the hosts.
- ◆ Use `bqueues -l` to check the queue thresholds, host limits, and user limits.
- ◆ If you submitted the jobs from a host that is the only host of its type in the cluster, by default, LSF tries to run the job on this host only. To run jobs on any available host type, submit the job with a command like:


```
% bsub -R "type==any" myjob
```

If none of these applies to your situation, contact support@platform.com.

7 LSF jobs go to a slow host while faster hosts are idle

Host processor speed is not the only factor in host selection:

- ◆ LSF uses the effective run queue lengths (displayed by `lsload -E`) to choose the least loaded execution hosts. Two hosts with the same effective run queue length are ordered according to their CPU factors.
- ◆ By default, LSF runs jobs on hosts of the same type as the submission host. To run jobs on any available host type, submit the job with a command like:


```
% bsub -R "type==any" myjob
```
- ◆ The faster hosts may not be included in the queue where the job was submitted.
- ◆ The faster hosts may not have the resources required by the job.
- ◆ The faster hosts may be closed or have closed dispatch windows or closed run windows.

If none of these applies to your situation, contact support@platform.com.

8 Application runs fine under UNIX or with `lshrun`, but fails or hangs when submitted through `bsub`

On some UNIX systems, certain applications only run with specific limit values. Different limit values or no limits can cause problems for these applications. `lshrun`, `lshgrun` and other interactive LSF commands copy the submission host environment, including any limits, to the execution host and `res` sets the submission environment on the execution host.

LSF Batch works differently. Jobs run in a queue, and are subject to queue limits, not submission host limits. By default, LSF Batch sets all limits to unlimited and only limits values explicitly set in the queue. To see the limit settings for your queues, use the command:

```
% bqueues -l -q queue_name
```

To troubleshoot this type of problem:

- 1 Run the application under UNIX to confirm that it works without LSF.
- 2 Create a small script like this:


```
#!/bin/sh
# display limits from command line
# Check the man pages for more limits that can be displayed
ulimit -Hc
ulimit -Hd
ulimit -Sc
ulimit -Sd
```
- 3 Run the script to determine the limits from the submission host under UNIX, and record the limit values.
- 4 Create a new version of the script, setting the limit values reported by the original script and running the command for your application. For example:


```
#!/bin/sh
# display limits from command line
# Check the man pages for more limits that can be displayed
ulimit -Hc 45333
ulimit -Hd 256
ulimit -Sc 24335
ulimit -Sd 256
<your application>
```

You can also set these limits in a queue in `LSB_CONFDIR/cluster_name/configdir/lsb.queues`.

Once the limit values are set correctly, your application should run fine under LSF Batch. If it still does not work, contact support@platform.com.

9 LSF Batch job runs in `/tmp` or cannot find home directory

The problem could be caused by:

- ◆ Different home directories existing for the same user account on the submission and execution hosts
- ◆ NFS automount problems cause LSF to look for incorrect mount points

- ◆ Permissions or ownership of your submission directory is incorrect for the home directory on the execution host
- ◆ You have a non-shared file system

A command may fail with the following error message due to a non-uniform file name space.

```
chdir(...) failed: no such file or directory
```

This error means that you are trying to run a command remotely, and:

- ◆ Your current working directory does not exist on the remote host
If your current working directory does not exist on a remote host, you should not execute commands remotely on that host.
or
- ◆ Your current working directory is mapped to a different name on the remote host.
If the directory exists, but is mapped to a different name on the remote host, you must create symbolic links to make them consistent.

LSF can resolve most, but not all, problems using automount. The automount maps must be managed through NIS.

10 Job is killed by some unknown signal or unknown reason

A job exit value smaller than 128 is the real exit value from your application.

If neither a user nor an LSF administrator kills the job, check if it was killed because it exceeded a resource usage limit:

- ◆ Use `bjobs -l` to check if any limit is specified on the job submission command.
- ◆ Use `bqueues -l` to check if any limit is specified on the queue.
- ◆ Use `bsub -q -I csh -c limit` to check if any other limit is set for the job.

If none of these applies to your situation, contact support@platform.com.

Index

A

- administration commands 18
- administrators 12
- automatic startup, configuring 38

B

- badadmin command, description 18
- batch configuration, checking 23
- batch configuration files 15
- batch daemon not responding, troubleshooting 58
- batch jobs
 - status 27
 - submitting 26
- batch system status, checking 23
- bsub command 26

C

- client-only hosts 12
- cluster
 - adding hosts 32
 - adding queues 36
 - reconfiguring 19
 - removing hosts 35
 - removing queues 36
 - starting 18
 - stopping 19
- cluster administrators 12
- cluster configuration, checking 21
- cluster configuration files 15
- cluster name 12
- cluster status, checking 22
- configuration
 - adding
 - hosts 32
 - queues 36
 - removing
 - hosts 35
 - queues 36
- cshrc.lsf file 18

D

- daemon log files 16
- daemons not responding, troubleshooting 58
- DEMO license, example 40
- dynamic load indexes 48
- dynamic shared resources
 - defining in ELIM 48
 - description 48
 - monitoring 49
 - using 50

E

- ELIM (external LIM)
 - defining dynamic shared resources 48
 - example 50
- error logs, location 54
- examples
 - DEMO license file 40
 - external LIM (ELIM) 50
 - lmstat command 45
 - permanent license file 41
- external LIM (ELIM)
 - defining dynamic shared resources 48
 - example 50

F

- FLEXlm server TCP port, setting in license.dat 44

H

- host types 12
- host-based resources 48
- hosts
 - adding 32
 - removing 35

I

- interactive tasks, running 28

J

- job execution, controlling 27
- job output 26
- job status 27
- jobs, submitting 26

L

- license file location 41
- license key, getting 42
- license problems, troubleshooting 55
- license server daemon, starting 42
- license status, checking 44
- licenses
 - examples 40
 - types 40
- LIM unavailable, troubleshooting 56
- lmstat command, example 45
- log files, daemon 16
- lost_and_found queue 36
- lsadmin command, description 18
- lsf.sudoers file, configuring 38
- lsrun command 26

M

master host 12

O

output, from jobs 26

P

permanent license
 description 40
 example 41
 installing 43
 updating 45
profile.lsf file 18

Q

queues
 adding 36
 lost_and_found 36
 removing 36

S

server daemons
 error logs 54
 troubleshooting 58
shared resources 48
startup, automatic 38