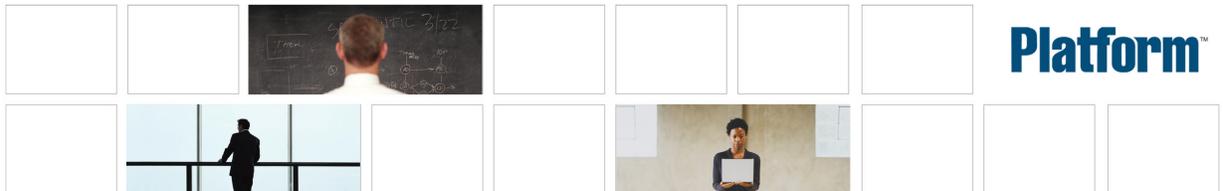

Release Notes for Platform LSF

Platform LSF
Version 7 Update 3
Release date: May 2008
Last modified: June 9, 2008



Contents

Release Notes for Platform LSF	3
Upgrade and Compatibility Notes	3
What's Changed in Platform LSF Version 7 Update 3	5
Known Issues	21
Download the Platform LSF Version 7 Distribution Packages	22
Install Platform LSF Version 7	24
Learn About Platform LSF Version 7	26
Get Technical Support	27
Copyright	28

Release Notes for Platform LSF

Release date: May 2008

Last modified: June 10, 2008

Comments to: doc@platform.com

Support: support@platform.com

Upgrade and Compatibility Notes

- Server host compatibility
- Upgrade from an earlier version of LSF on UNIX and Linux
- Migrate your existing LSF Version 7 cluster to Version 7 Update 3 on UNIX and Linux
- Migrate LSF on Windows
- Maintenance pack and update availability
- System requirements
- API compatibility

What's new in Platform LSF Version 7 Update 3

For detailed information about what's new in Platform LSF Version 7 Update 3, visit the Platform Computing Web site to see [Features, Benefits & What's New](#).

Server host compatibility

Important:

To use new features introduced in Platform LSF Version 7, you *must* upgrade all hosts in your cluster to LSF 7.

LSF 6.x and 5.x servers are compatible with Platform LSF Version 7 master hosts. All LSF 6.x and 5.x features are supported by LSF 7 master hosts.

Upgrade from an earlier version of LSF on UNIX and Linux

Follow the steps in *Upgrading Platform LSF on UNIX and Linux* (lsf_upgrade_unix.pdf) to run `lsfinstall` to *upgrade* LSF:

- Upgrade a pre-version 7 UNIX or Linux cluster to LSF Version 7 Update 3
- Upgrade an LSF Version 7 Update 2 UNIX or Linux cluster to LSF Version 7 Update 3

Important:

DO NOT use the UNIX and Linux upgrade steps to migrate an existing LSF 7 cluster or LSF 7 Update 1 cluster to LSF 7 Update 3. Follow the manual steps in the document *Migrating to Platform LSF Version 7 Update 3 on UNIX and Linux* to migrate an existing LSF 7 Update 1 cluster to LSF 7 Update 3 on UNIX and Linux.

Migrate your existing LSF 7 cluster to Update 3 on UNIX and Linux

Follow the steps in *Migrating to Platform LSF Version 7 Update 3 on UNIX and Linux* (lsf_migrate_unix.pdf) to migrate an *existing* LSF 7 cluster:

- Migrate an existing LSF Version 7 cluster to LSF 7 Update 3 on UNIX and Linux
- Migrate an existing LSF 7 Update 1 cluster to LSF 7 Update 3 on UNIX and Linux

Note:

DO NOT use these steps to migrate an existing LSF 7 Update 2 cluster to LSF 7 Update 3. Follow the steps in *Upgrading Platform LSF on UNIX and Linux* to upgrade LSF.

Migrate LSF on Windows from an earlier version

To migrate a *pre-version* 7 cluster to a new LSF 7 on Windows cluster, follow the steps in *Migrating Your Windows Cluster to Platform LSF Version 7* (lsf_migrate_windows.pdf).

Note:

DO NOT use these steps to migrate an existing LSF 7 cluster to LSF 7 Update 3.

Migrate your existing LSF cluster to Update 3 on Windows

To migrate an *existing* LSF 7 Windows cluster to LSF 7 Update 3 on Windows, follow the steps in *Migrating Platform LSF Version 7 to Update 3 on Windows* (lsf_migrate_windows_to_update3.pdf).

Note:

DO NOT use these steps to migrate a pre-version 7 cluster to LSF 7 Update 3.

Maintenance pack and update availability

At release, Platform LSF Version 7 Update 3 includes all bug fixes and solutions up to and including bug fixes and solutions up to April 11, 2008. Fixes after that date will be included in the next LSF update.

System requirements

See the Platform Computing Web site for information about supported operating systems and system requirements for [Platform LSF](#).

API compatibility

Full backward compatibility: your applications will run under LSF Version 7 without changing any code.

The Platform LSF Version 7 API is fully compatible with the LSF Version 6.x and 5.x APIs. An application linked with the LSF Version 6.x or 5.x libraries will run under LSF Version 7 without relinking.

To take full advantage of new Platform LSF Version 7 features, you should recompile your existing LSF applications with LSF Version 7.

New and changed LSF APIs

See the *LSF API Reference* for more information.

The `lsb_modreservation()` API has been added for LSF Version 7 Update 3 to modify advance reservations.

The following APIs have changed for LSF Version 7 Update 3:

- `lsb_geteventrec()`, `lsb_puteventrec()` — add the `requeueEValues` field to the `jobNewLog` and `jobModLog` data structures
- `lsb_modify()`, `lsb_readjobinfo()`, `lsb_readjobinfo_cond()`, `lsb_submit()`, `lsb_submitframe()` — add the `requeueEValues` field to the submit data structure
- `lsb_geteventrec()` and `lsb_puteventrec()` — add the `initChkpntPeriod` and `migThreshold` fields to the `jobNewLog` and the `jobModLog` data structures
- `lsb_jsdl2submit()`, `lsb_modify()`, `lsb_readjobinfo()`, `lsb_readjobinfo_cond()`, `lsb_submit()`, `lsb_submitframe()` — add the `initChkpntPeriod` and `migThreshold` fields to the submit data structure
- `lsb_geteventrec()`, `lsb_puteventrec()` — add the `jgrpNewLog` data structure
- `lsb_addreservation()` — changes interface of the `addRsvRequest` data structure
- `lsb_reservationinfo()` — changes interface of the `rsvInfoEnt` data structure

What's Changed in Platform LSF Version 7 Update 3

- Changed behavior
- New and changed configuration parameters and environment variables
- New commands
- Changed commands, options, and output
- New and changed files
- New and changed accounting and job event fields
- Bugs fixed since November 2007

Changed behavior

Platform LSF Session Scheduler

Platform LSF Session Scheduler addresses the need for efficient scheduling and dispatching large volumes of independent jobs with short run times. Platform LSF Session Scheduler is a separate add-on product like LSF License Scheduler and LSF MultiCluster.

As clusters grow and the volume of workload increases, the need to delegate scheduling decisions increases. While traditional Platform LSF job submission, scheduling, and dispatch methods such as job arrays or job chunking are well suited to a mix of long and short running jobs, or jobs with dependencies on each other, Session Scheduler improves throughput and performance of the LSF scheduler by enabling multiple tasks to be submitted as a single LSF job.

Session Scheduler implements a hierarchal, personal scheduling paradigm that provides very low-latency execution. With very low latency per job, Session Scheduler is ideal for executing very short jobs, whether they are a list of tasks, or job arrays with parametric execution.

Each Session Scheduler job is dynamically scheduled in a similar manner to a parallel job. Each instance of the `ssched` command then manages its own workload within its assigned allocation. Work is submitted as a task array or a task definition file.

See *Installing and Running Platform LSF Session Scheduler* for more information.

Automatic job group cleanup

When an implicitly created job group becomes empty, it can be automatically deleted by LSF. Job groups that can be automatically deleted cannot:

- Have limits specified including their child groups
- Have explicitly created child job groups
- Be attached to any SLA

Configure `JOB_GROUP_CLEAN=Y` in `l sb. params` to enable automatic job group deletion.

Automatic job group deletion does not delete job groups attached to SLA service classes. Use `bgdel` to manually delete job groups attached to SLAs.

How job groups are created

Job groups can be created *explicitly* or *implicitly*:

- A job group is created *explicitly* with the `bgadd` command.
- A job group is created *implicitly* by the `bsub -g` or `bmod -g` command when the specified group does not exist. Job groups are also created implicitly when a default job group is configured (`DEFAULT_JOBGROUP` in `l sb. params` or `LSB_DEFAULT_JOBGROUP` environment variable).

Job groups created when jobs are attached to an SLA service class at submission are implicit job groups (`bsub -sla service_class_name -g job_group_name`). Job groups attached to an SLA service class with `bgadd` are explicit job groups (`bgadd -sla service_class_name job_group_name`).

The `GRP_ADD` event in `l sb. events` indicates how the job group was created:

- `0x01` - job group was created explicitly
- `0x02` - job group was created implicitly

For example:

```
GRP_ADD "7.02" 1193032735 1285 1193032735 0 "/Z" "" "user1" "" "" 2 0 "" -1
1
```

means job group `/Z` is an explicitly created job group.

Child groups can be created explicitly or implicitly under any job group. Only an implicitly created job group which has no limit and is not attached to any SLA can be automatically deleted once it becomes empty. An empty job group is a job group that has no jobs associated with it (including finished jobs). `NJOBS` displayed by `bjgroup` is 0.

Use `brsvmod` to modify advance reservations

Use `brsvmod` to make the following changes to an existing advance reservation:

- Modify start time (postpone or move closer)
- Modify the duration of the reservation window (and thus the end time)
- Modify the slot numbers required by the reservation (add or remove slots with hosts)

- Modify the host or hostgroup list (add or remove hosts or hostgroups)
- Modify the user or user group
- Add hosts by resource requirement (-R)
- Modify the reservation type (open or closed)
- Disable the specified occurrences of a recurring reservation

Use `brsvmod` to shift, extend or reduce the time window horizontally; grow or shrink the size vertically.

Specify successful application exit values.

Jobs that exit with one of the exit codes specified by `SUCCESS_EXIT_VALUES` in an application profile are marked as `DONE`. These exit values are not be counted in the `EXIT_RATE` calculation.

0 always indicates application success regardless of `SUCCESS_EXIT_VALUES`.

If both `SUCCESS_EXIT_VALUES` and `REQUEUE_EXIT_VALUES` are defined, job will be set to `PEND` state and requeued.

`SUCCESS_EXIT_VALUES` has no effect on pre-exec and post-exec commands. The value is only used for user jobs.

If the job exit value falls into `SUCCESS_EXIT_VALUES`, the job will be marked as `DONE`. Job dependencies on done jobs behave normally.

For parallel jobs, the exit status refers to the job exit status and not the exit status of individual tasks.

Exit codes for jobs terminated by LSF are excluded from success exit value even if they are specified in `SUCCESS_EXIT_VALUES`.

For example, if `SUCCESS_EXIT_VALUES=2` is defined, jobs exiting with 2 are marked as `DONE`. However, if LSF cannot find the current working directory, LSF terminates the job with exit code 2, and the job is marked as `EXIT`. The appropriate termination reason is displayed by `bacct`.

Job-level success exit values specified with the `LSB_SUCCESS_EXIT_VALUES` environment variable override application profile level specification of `SUCCESS_EXIT_VALUES` in `l sb. appl i cat i ons`.

Control how many times LSF retries pre-execution commands

By default, if job pre-execution fails, LSF retries the job automatically.

Configure `MAX_PREEEXEC_RETRY` to limit the number of times LSF retries job pre-execution. Pre-execution retry is configured cluster-wide (`l sb. params`), at the queue level (`l sb. queues`), and at the application level (`l sb. appl i cat i ons`).

`MAX_PREEEXEC_RETRY` in `l sb. appl i cat i ons` overrides `l sb. queues`, and `l sb. queues` overrides `l sb. params` configuration.

Control how many times a job can be requeued

By default, if a job fails and its exit value falls into `REQUEUE_EXIT_VALUES`, LSF requeues the job automatically. Jobs that fail repeatedly are requeued without limit, which can result in reduced performance and throughput.

To to limit the number of times a failed job is requeued, set `MAX_JOB_REQUEUE` cluster wide (1 sb. params), in the queue definition (1 sb. queues), or in an application profile (1 sb. applications).

Specify an integer greater than zero (0).

`MAX_JOB_REQUEUE` in 1 sb. applications overrides 1 sb. queues, and 1 sb. queues overrides 1 sb. params configuration.

When `MAX_JOB_REQUEUE` is set, if a job fails and its exit value falls into `REQUEUE_EXIT_VALUES`, the number of times the job has been requeued is increased by 1 and the job is requeued. When the requeue limit is reached, the job is suspended with `PSUSP` status. If a job fails and its exit value is not specified in `REQUEUE_EXIT_VALUES`, the job is not requeued.

Automatic job requeue

The reserved keyword `all` specifies all exit codes. Exit codes are typically between 0 and 255. Use a tilde (~) to exclude specified exit codes from the list.

For example:

```
REQUEUE_EXIT_VALUES=all ~1 ~2 EXCLUDE(9)
```

Jobs exited with all exit codes except 1 and 2 are requeued. Jobs with exit code 9 are requeued so that the failed job is not rerun on the same host (exclusive job requeue).

Job-level automatic requeue

Use `bsub -Q` to submit a job that is automatically requeued if it exits with the specified exit values. Use spaces to separate multiple exit codes. The reserved keyword `all` specifies all exit codes. Exit codes are typically between 0 and 255. Use a tilde (~) to exclude specified exit codes from the list.

Job-level requeue exit values override application-level and queue-level configuration of the parameter `REQUEUE_EXIT_VALUES`, if defined.

Jobs running with the specified exit code share the same application and queue with other jobs.

For example:

```
bsub -Q "all ~1 ~2 EXCLUDE(9)" myjob
```

Jobs exited with all exit codes except 1 and 2 are requeued. Jobs with exit code 9 are requeued as exclusive jobs.

Precendence of checkpointing options

If checkpoint-related configuration is specified in both the queue and an application profile, the application profile setting overrides queue level configuration.

If checkpoint-related configuration is specified in the queue, application profile, and at job level:

- Application-level and job-level parameters are merged. If the same parameter is defined at both job-level and in the application profile, the job-level value overrides the application profile value.
- The merged result of job-level and application profile settings override queue-level configuration.

Checkpointing MultiCluster jobs

To enable checkpointing of MultiCluster jobs, define a checkpoint directory in both the send-jobs and receive-jobs queues (CHKPNT in `lsf.queues`), or in an application profile (CHKPNT_DIR, CHKPNT_PERIOD, CHKPNT_INITPERIOD, CHKPNT_METHOD in `lsf.applications`) of both submission cluster and execution cluster. LSF uses the directory specified in the execution cluster.

Checkpointing is not supported if a job runs on a leased host.

Defining consumable resources

Specify resources as consumable in the CONSUMABLE column of the RESOURCE section of `lsf.shared` to explicitly control if a resource is consumable. Static and dynamic numeric resources can be specified as consumable. CONSUMABLE is optional. The defaults for the consumable attribute are:

- Built-in indicies:
 - The following are consumable: r15s, r1m, r15m, ut, pg, io, ls, it, tmp, swp, mem.
 - All other built-in static resources are not consumable. (e.g., ncpus, ndisks, maxmem, maxswp, maxtmp, cpuf, type, model, status, rexpri, server, hname).
- External shared resources:
 - All numeric resources are consumable.
 - String and boolean resources are not consumable.

You should only specify consumable resources in the rusage section of a resource requirement string. Non-consumable resources are ignored in rusage sections.

A non-consumable resource should not be releasable. Non-consumable numeric resource should be able to used in order, select and same sections of a resource requirement string.

When `LSF_STRICT_RESREQ=Y` in `lsf.conf`, LSF rejects resource requirement strings where an rusage section contains a non-consumable resource.

Viewing consumable resources

Use `lsinfo -l` to view consumable resources. For example:

```

lsinfo -l switch
RESOURCE_NAME:  switch
DESCRIPTION:    Network Switch
TYPE            ORDER  INTERVAL  BUILTIN  DYNAMIC  RELEASE  CONSUMABLE
Numeric        Inc    0         No       No       No       No
lsinfo -l specman
RESOURCE_NAME:  specman
DESCRIPTION:    Specman
TYPE            ORDER  INTERVAL  BUILTIN  DYNAMIC  RELEASE  CONSUMABLE
Numeric        Dec    0         No       No       Yes      Yes

```

Static resources reported by LIM

The resources `ncpus`, `nprocs`, `ncores`, `nthreads`, `maxmem`, `maxswp`, and `maxtmp` are not static on UNIX hosts that support dynamic hardware reconfiguration. LIM now reports `nprocs`, `ncores` and `nthreads`. Use `lsinfo` to list the resources available in your cluster.

Index	Measures	Units	Determined by
nprocs	number of physical processors	processors	LIM

Index	Measures	Units	Determined by
ncores	number of cores per physical processor	cores	LIM
nthreads	number of threads per processor core	threads	LIM

How environment variables determine elim hosts

If you use the default keyword for any external resource in `lsf.cluster.cluster_name`, all `elim` executables in `LSF_SERVERDIR` run on all hosts in the cluster. You can control the hosts on which your `elim` executables run by using the environment variables `LSF_MASTER`, `LSF_RESOURCES`, and `ELIM_ABORT_VALUE`. These environment variables provide a way to ensure that `elim` executables run only when they are programmed to report the values for resources expected on a host.

- `LSF_MASTER`—You can program your `elim` to check the value of the `LSF_MASTER` environment variable. The value is `Y` on the master host and `N` on all other hosts. An `elim` executable can use this parameter to check the host on which the `elim` is currently running.
- `LSF_RESOURCES`—When the LIM starts an MELIM on a host, the LIM checks the resource mapping defined in the ResourceMap section of `lsf.cluster.cluster_name`. Based on the mapping (default, all, or a host list), the LIM sets `LSF_RESOURCES` to the list of resources expected on the host. Use `LSF_RESOURCES` in a checking header to verify that an `elim` is programmed to collect values for at least one of the resources listed in `LSF_RESOURCES`.
- `ELIM_ABORT_VALUE`—An `elim` should exit with `ELIM_ABORT_VALUE` if the `elim` is not programmed to collect values for at least one of the resources listed in `LSF_RESOURCES`. The MELIM does not restart an `elim` that exits with `ELIM_ABORT_VALUE`.

Enhanced resource requirement parsing (BETA)

Platform LSF Version 7 Update 3 improves the performance and rigor of resource requirement select string syntax. The enhancement is available for BETA evaluation only. Contact lsfbeta@platform.com for more information.

Strict resource requirement syntax is enabled by configuration parameter (`LSF_STRICT_RESREQ=Y` in `lsf.conf`). This enhancement only affects `select[]` resource requirement strings. The enhancement does not affect other resource requirement sections (`rusage[]`, `order[]`, `span[]`, `same[]`).

Command and parameter syntax

Syntax for Platform LSF commands and parameters has been clarified.

Syntax rules

bold

Type the text exactly as shown.

user logoff

italics

The argument is a variable, which must be replaced with a real value you provide.

job_ID

Argument in square brackets []

The argument surrounded by square brackets is optional. Do not enter the brackets.

user list [-l]

Argument in brace brackets { }

One of the arguments surrounded by brace brackets is required. You cannot use both options together. Do not enter the brackets.

user list {-s | -n processors}

OR bars |

OR bars separate items in a list. You can only enter one of the items in the list. Do not enter the bar.

config [-h | -V]

Quotes " or '

Double or single quotes must be entered exactly as shown.

"job_ID[index_list]"

Commas ,

Commas must be entered exactly as shown.

-C time0,time1

Ellipsis following an argument ...

You can repeat the item that precedes the ellipsis. You must separate items with a space. Do not enter the ellipsis.

host_name ...

Comma followed by ellipsis ,...

You can repeat the item that precedes the ellipsis. You must separate items with a comma. Do not enter the ellipsis.

host_name[,host_name...]

Parentheses ()

Parentheses must be entered exactly as shown.

EXCLUDE(exit_code ...)

Subcommand syntax

Commands that have subcommands follow the same syntax rules, but have an additional layer of complexity. The syntax for commands that have subcommands appears as follows:

command

command subcommand [options]

command -h | -V

command

The command can be issued without any options, usually to open a command shell. From within this command shell, subcommands can be issued without prefacing them with the command, until such time as the command shell is closed. The command can also be issued with subcommands and options in a single string, to open the command shell, perform the action, and close the command shell in a single command string.

subcommand

This is a list of subcommands that can be entered from within the open command shell, or can be prefaced by the command, to open the command shell, perform the action, and close the command shell in a single command string.

options

These are the options that apply to the subcommand. The basic syntax rules apply.

New and changed configuration parameters and environment variables

The following configuration parameters and environment variables are new or changed for LSF Version 7 Update 3:

lsb.applications

- **CHKPNT_DIR=chkpnt_dir**—Specifies the checkpoint directory for automatic checkpointing for the application. To enable automatic checkpoint for the application profile, administrators must specify a checkpoint directory in the configuration of the application profile. If CHKPNT_PERIOD, CHKPNT_INITPERIOD or CHKPNT_METHOD was set in an application profile but CHKPNT_DIR was not set, a warning message is issued and those settings are ignored.
- **CHKPNT_INITPERIOD=init_chkpnt_period**—Specifies the initial checkpoint period in minutes. CHKPNT_DIR must be set in the application profile for this parameter to take effect. The periodic checkpoint specified by CHKPNT_PERIOD does not happen until the initial period has elapse. Specify a positive integer. Job-level command line values override the application profile configuration.
- **CHKPNT_PERIOD=chkpnt_period**—Specifies the checkpoint period for the application in minutes. CHKPNT_DIR must be set in the application profile for this parameter to take effect. The running job is checkpointed automatically every checkpoint period. Specify a positive integer.
- **CHKPNT_METHOD=chkpnt_method**—Specifies the checkpoint method. CHKPNT_DIR must be set in the application profile for this parameter to take effect. Job-level command line values override the application profile configuration.
- **MAX_JOB_PREEMPT=integer**—The maximum number of times a job can be preempted. Applies to queue-level jobs only.
- **MAX_JOB_REQUEUE=integer**
The maximum number of times to requeue a job automatically.
- **MIG=minutes**—Enables automatic job migration and specifies the migration threshold for checkpointable or rerunnable jobs, in minutes. LSF automatically migrates jobs that have been in the SSUSP state for more than the specified number of minutes. A value of 0 specifies that a suspended job is migrated immediately. The migration threshold applies

to all jobs running on the host. Job-level command line migration threshold overrides threshold configuration in application profile and queue. Application profile configuration overrides queue level configuration.

- **NO_PREEMPT_FINISH_TIME=*minutes* | *percentage***—Prevents preemption of jobs that will finish within the specified number of minutes or the specified percentage of the estimated run time or run limit. Specifies that jobs due to finish within the specified number of minutes or percentage of job duration should not be preempted, where *minutes* is wall-clock time, not normalized time. Percentage must be greater than 0 or less than 100% (between 1% and 99%).
- **NO_PREEMPT_RUN_TIME=*minutes* | *percentage***—Prevents preemption of jobs that have been running for the specified number of minutes or the specified percentage of the estimated run time or run limit. Specifies that jobs that have been running for the specified number of minutes or longer should not be preempted, where *minutes* is wall-clock time, not normalized time. Percentage must be greater than 0 or less than 100% (between 1% and 99%).
- **REQUEUE_EXIT_VALUES=[*exit_code* ...] [EXCLUDE(*exit_code* ...)]**—Enables automatic job requeue and sets the LSB_EXIT_REQUEUE environment variable. Use spaces to separate multiple exit code values. Application-level exit values override queue-level values. Job-level exit values (bsub -Q) override application-level and queue-level values. *exit_code* has the following form:

```
" [all] | [~number ...] | [number ...] "
```

The reserved keyword all specifies all exit codes. Exit codes are typically between 0 and 255. Use a tilde (~) to exclude specified exit codes from the list.

- **SUCCESS_EXIT_VALUES=[*exit_code* ...]**—Specifies exit values used by LSF to determine if job was done successfully. Use spaces to separate multiple exit codes. Job-level user-defined success exit values specified with the LSB_SUCCESS_EXIT_VALUES environment variable override the configuration in application profile. Use SUCCESS_EXIT_VALUES for applications that successfully exit with non-zero values so that LSF does not interpret non-zero exit codes as job failure. *exit_code* should be the value between 0 and 255. Use spaces to separate exit code values.
- When LSF_STRICT_RESREQ=Y in lsf.conf, LSF rejects resource requirement strings where an rusage section contains a non-consumable resource

lsb.params

- **JOB_GROUP_CLEAN=Y | N**—If JOB_GROUP_CLEAN = Y, implicitly created job groups that are empty and have no limits assigned to them are automatically deleted.
- **MAX_JOB_PREEMPT=*integer***—The maximum number of times a job can be preempted. Applies to queue-level jobs only.
- **MAX_JOB_REQUEUE=*integer***

The maximum number of times to requeue a job automatically.

- **NO_PREEMPT_FINISH_TIME=*minutes* | *percentage***—Prevents preemption of jobs that will finish within the specified number of minutes or the specified percentage of the estimated run time or run limit. Specifies that jobs due to finish within the specified number of minutes or percentage of job duration should not be preempted, where *minutes* is wall-clock time, not normalized time. Percentage must be greater than 0 or less than 100% (between 1% and 99%).
- **NO_PREEMPT_RUN_TIME=*minutes* | *percentage***—Prevents preemption of jobs that have been running for the specified number of minutes or the specified percentage of the estimated run time or run limit. Specifies that jobs that have been running for the specified

number of minutes or longer should not be preempted, where *minutes* is wall-clock time, not normalized time. Percentage must be greater than 0 or less than 100% (between 1% and 99%).

lsb.queues

- **CHKPNT=*chkpnt_dir* [*chkpnt_period*]**—If checkpoint-related configuration is specified in both the queue and an application profile, the application profile setting overrides queue level configuration. If checkpoint-related configuration is specified in the queue, application profile, and at job level:
 - Application-level and job-level parameters are merged. If the same parameter is defined at both job-level and in the application profile, the job-level value overrides the application profile value.
 - The merged result of job-level and application profile settings override queue-level configuration.

To enable checkpointing of MultiCluster jobs, define a checkpoint directory in both the send-jobs and receive-jobs queues (CHKPNT in `lsb.queues`), or in an application profile (CHKPNT_DIR, CHKPNT_PERIOD, CHKPNT_INITPERIOD, CHKPNT_METHOD in `lsb.applications`) of both submission cluster and execution cluster. LSF uses the directory specified in the execution cluster.

- **MAX_JOB_PREEMPT=*integer***—The maximum number of times a job can be preempted. Applies to queue-level jobs only.
- **MAX_JOB_REQUEUE=*integer***

The maximum number of times to requeue a job automatically.

- **MIG=*minutes***—Job-level command line migration threshold overrides threshold configuration in application profile and queue. Application profile configuration overrides queue level configuration. When a host migration threshold is specified, and is lower than the value for the job, the queue, or the application, the host value is used..
- **REQUEUE_EXIT_VALUES=[*exit_code*...] [EXCLUDE(*exit_code*...)]**—Enables automatic job requeue and sets the LSB_EXIT_REQUEUE environment variable. Use spaces to separate multiple exit codes. Application-level exit values override queue-level values. Job-level exit values (bsub -Q) override application-level and queue-level values. *exit_code* has the following form:

```
"[all] [~number ...] | [number ...]"
```

The reserved keyword all specifies all exit codes. Exit codes are typically between 0 and 255. Use a tilde (~) to exclude specified exit codes from the list.

- When LSF_STRICT_RESREQ=Y in `lsf.conf`, LSF rejects resource requirement strings where a resource section contains a non-consumable resource

lsf.cluster

- RESOURCES in Host section lists all static Boolean resources and static or dynamic numeric and string resources available on this host.
- ELIMARGS=*cmd_line_args* in Parameters section specifies command-line arguments required by an elim executable on startup. Used only when the external load indices feature is enabled.
- ELIM_POLL_INTERVAL=*seconds*—in Parameters section specifies a time interval, in seconds, that the LIM samples external load index information. If your elim executable

is programmed to report values more frequently than every 5 seconds, set the `ELIM_POLL_INTERVAL` so that it samples information at a corresponding rate.

- `LSF_ELIM_BLOCKTIME=seconds`—in Parameters section. UNIX only; used when the external load indices feature is enabled. Maximum amount of time the master external load information manager (MELIM) waits for a complete load update string from an `elim` executable. After the time period specified by `LSF_ELIM_BLOCKTIME`, the MELIM writes the last string sent by an `elim` in the LIM log file (`lim.log`, `host_name`) and restarts the `elim`. Defining `LSF_ELIM_BLOCKTIME` also triggers the MELIM to restart `elim` executables if the `elim` does not write a complete load update string within the time specified for `LSF_ELIM_BLOCKTIME`.
- `LSF_ELIM_DEBUG=y`—in Parameters section. UNIX only; used when the external load indices feature is enabled. When this parameter is set to `y`, all external load information received by the load information manager (LIM) from the master external load information manager (MELIM) is logged in the LIM log file (`lim.log`, `host_name`). Defining `LSF_ELIM_DEBUG` also triggers the MELIM to restart `elim` executables if the `elim` does not write a complete load update string within the time specified for `LSF_ELIM_BLOCKTIME`.
- `LSF_ELIM_RESTARTS=integer`—in Parameters section. UNIX only; used when the external load indices feature is enabled. Maximum number of times the master external load information manager (MELIM) can restart `elim` executables on a host. Defining this parameter prevents an ongoing restart loop in the case of a faulty `elim`. The MELIM waits the `LSF_ELIM_BLOCKTIME` to receive a complete load update string before restarting the `elim`. The MELIM does not restart any `elim` executables that exit with `ELIM_ABORT_VALUE`.

Important:

Either `LSF_ELIM_BLOCKTIME` or `LSF_ELIM_DEBUG` must also be defined; defining these parameters triggers the MELIM to restart `elim` executables.

lsf.conf

- `LSF_PAM_CLEAN_JOB_DELAY=time_seconds`—The number of seconds LSF waits before killing a parallel job with failed tasks. Specifying `LSF_PAM_CLEAN_JOB_DELAY` implies that if any parallel tasks fail, the entire job should exit without running the other tasks in the job. The job is killed if any task exits with a non-zero exit code. Specify a value greater than or equal to zero (0). Applies only to PAM jobs.
- `LSB_DEBUG_CMD` adds `LC_ADVRSV` class to log advance reservation modifications with `brsvmod`.
- `EGO_PREDEFINED_RESOURCES`—When Platform EGO is enabled in the LSF cluster (`LSF_ENABLE_EGO=Y`), you also can set the several EGO parameters related to LIM, PIM, and ELIM in either `lsf.conf` or `ego.conf`. All clusters must have the same value of `EGO_PREDEFINED_RESOURCES` in `lsf.conf` to enable the `nprocs`, `ncores`, and `nthreads` host resources in remote clusters to be usable.

lsf.shared

A resource name cannot be any of the following reserved keywords:

```
cpu cpuf io logins ls idle maxmem maxswp maxtmp type model
status it mem ncpus nprocs ncores nthreads
define_ncpus_cores define_ncpus_procs define_ncpus_threads
ndisks pg r15m r15s r1m swap swp tmp ut
```

CONSUMABLE in Resource section—Explicitly controls if a resource is consumable. Applies to static or dynamic numeric resources. Static and dynamic numeric resources can be specified as consumable. CONSUMABLE is optional. The defaults for the consumable attribute are:

- Built-in indicies:
 - The following are consumable: r15s, r1m, r15m, ut, pg, io, ls, it, tmp, swp, mem.
 - All other built-in static resources are not consumable. (e.g., ncpus, ndisks, maxmem, maxswp, maxtmp, cpuf, type, model, status, rexpri, server, hname).
- External shared resources:
 - All numeric resources are consumable.
 - String and boolean resources are not consumable.

You should only specify consumable resources in the rusage section of a resource requirement string. Non-consumable resources are ignored in rusage sections. A non-consumable resource should not be releasable. Non-consumable numeric resource should be able to used in order, select and same sections of a resource requirement string.

When `LSF_STRICT_RESREQ=Y` in `lsf.conf`, LSF rejects resource requirement strings where an rusage section contains a non-consumable resource.

Environment variables

- **ELIM_ABORT_VALUE**—Used when writing an `elim` executable to test whether the `elim` should run on a particular host. If the host does not have or share any of the resources listed in the environment variable `LSF_RESOURCES`, your `elim` should exit with `$ELIM_ABORT_VALUE`. When the MELIM finds an `elim` that exited with `ELIM_ABORT_VALUE`, the MELIM marks the `elim` and does not restart it on that host.
- **LSB_SUCCESS_EXIT_VALUES**=[*exit_code ...*] — Specifies the exit values that indicate successful execution for applications that successfully exit with non-zero values. Use spaces to separate multiple exit codes. *exit_code* should be the value between 0 and 255. User-defined job-level `LSB_SUCCESS_EXIT_VALUES` overrides application profile level specification of `SUCCESS_EXIT_VALUES` in `lsb.appl icat ions`.
- **LSF_MASTER**—Set by the LIM to identify the master host. The value is Y on the master host and N on all other hosts. An `elim` executable can use this parameter to check the host on which the `elim` is currently running.
- **LSF_RESOURCES**=*dynamic_external_resource_name...*—A space-separated list of dynamic external resources. When the LIM starts a master external load information manager (MELIM) on a host, the LIM checks the resource mapping defined in the ResourceMap section of `lsf.cluster.cluster_name`. Based on the mapping (default, all, or a host list), the LIM sets `LSF_RESOURCES` to the list of resources expected on the host and passes the information to the MELIM. Used when the external load indices feature is enabled.

New commands

The following new commands have been added to LSF Version 7 Update 3:

brsvmod

Modifies an advance reservation. `brsvmod` replaces advance reservation option values previously created, extends or reduces the reservation time window, or adds or removes reserved hosts of the advance reservation specified by *reservation_ID*. For a recurring reservation, can disable specified occurrences of the reservation.

perfadmin

Starts or stops the PERF services, or shows status to administer the LSF Reports (PERF) services. Run the command on the PERF host to control the following PERF services: loader controller (`plc`), job data transformer (`j obdt`), and data purger (`purger`). Run the command on the Derby database host to control the Derby database service (`derbydb`). If PERF services are controlled by EGO, let the EGO service controller start and stop the PERF services. `perfadmin` can only be used by LSF administrators.

perfremoverc

Prevents automatic startup of PERF daemons on a UNIX host when a system reboot command is issued. After this script/command is issued, PERF daemons no longer start automatically if the host gets rebooted. In such a case, you must manually start daemons after the host has started up. You must be logged on as `root` to run `perfremoverc`.

perfsetrc

Configures a UNIX host to allow automatic startup of PERF daemons on the machine when a system reboot command is issued. Creates the file `perf` under the system startup directory. For ease of administration, you should enable automatic startup. This starts PERF daemons automatically when the host restarts. If you do not configure hosts to start automatically, PERF daemons must be started manually. You must be logged on as `root` to run `perfsetrc`.

pmcadmin

`pmcadmin` administers the Platform Management Console (PMC). Always run this command on the host that runs PMC. If PMC services are controlled by EGO, let the EGO service controller start and stop the PMC services. `pmcadmin` can only be used by LSF administrators.

ssacct

Displays accounting statistics about finished LSF jobs run through Platform LSF Session Scheduler. By default, displays accounting statistics for all finished jobs submitted by the user who invoked the command.

ssched

Submit tasks through Platform LSF Session Scheduler. Options can be specified on the `ssched` command line or on a line in a task definition file. If specified on the command line, the option applies to all tasks, whether specified on the command line or in a file. Options specified in a file apply only to the command on that line. Options in the task definition file override the same option specified on the command line.

Changed commands, options, and output

The following command options and output are new or changed for LSF Version 7 Update 3:

bacct -U

The `-U` option now displays historical information about reservation modifications.

badmin

-c *class_name* adds LC_ADVRSV class to log advance reservation modifications with brsvmod.

bapp -l

Displays new application profile configuration parameters for checkpoint-related settings:

- CHKPNT_DIR—The checkpoint directory, if automatic checkpointing is enabled for the application profile.
- CHKPNT_INITPERIOD—The initial checkpoint period in minutes. The periodic checkpoint does not happen until the initial period has elapsed.
- CHKPNT_PERIOD—The checkpoint period in minutes. The running job is checkpointed automatically every checkpoint period.
- CHKPNT_METHOD—The checkpoint method.

bgdel

- bgdel 0—allows normal users and admins to delete the job groups they created
- bgdel -u *username* 0—allows administrators to delete the job groups created by the specified user
- bgdel -u all 0—allows administrators to delete all empty job groups for all users.
- bgdel -c *job_group_name*—allows users to delete all the empty groups below the requested *job_group_name* including the *job_group_name* itself.

bhist -l

Displays new job submission information for initial checkpoint period (bsub -k i n i t) and job migration threshold (bsub -m i g)

bhosts -s

bhost s -s only shows consumable resources.

bjobs -l

Displays new job submission information for initial checkpoint period (bsub -k i n i t) and job migration threshold (bsub -m i g)

bmod

- -Q "[*exit_code* ...] [EXCLUDE(*exit_code* ...)]" — modifies submitted automatic job requeue exit values. -Q does not affect running jobs. For rerunnable and requeue jobs, -Q affects the next run.
- -mig *migration_threshold* | -mign — modifies the submitted job migration threshold for checkpointable or rerunnable jobs in minutes.
- -k "*checkpoint_dir* [init=*initial_checkpoint_period*] [*checkpoint_period*]" | -kn — init modifies the initial checkpoint period in minutes.

brsvadd

- -d "*description*" specifies a description for the reservation to be created. The description must be provided as a double quoted text string. The maximum length is 512 characters.

- `-N reservation_name` specifies a user-defined advance reservation name unique in an LSF cluster. The name is a string of letters, numeric characters, underscores, and dashes beginning with a letter. The maximum length of the name is 39 characters.

brsvs

`-z all | "host_name"` shows a planner with only the weekly items that have reservation configurations displayed. Empty lines are omitted.

bsub

`-R res_req` displays hosts with the specified resource requirements. When `LSF_STRICT_RESREQ=Y` in `lsf.conf`, LSF rejects resource requirement strings where a rusage section contains a non-consumable resource.

`bsub` adds the following new options:

`-k "checkpoint_dir [init=initial_checkpoint_period] [checkpoint_period] [method=method_name]"`

`-k init` optionally, specifies an initial checkpoint period in minutes. Specify a positive integer. The first checkpoint does not happen until the initial period has elapsed. After the first checkpoint, the job checkpoint frequency is controlled by the normal job checkpoint interval.

`-mig migration_threshold`

Specifies the migration threshold for checkpointable or rerunnable jobs in minutes. Enables automatic job migration and specifies the migration threshold, in minutes. A value of 0 (zero) specifies that a suspended job should be migrated immediately.

Command-level job migration threshold overrides application profile and queue-level settings.

Where a host migration threshold is also specified, and is lower than the job value, the host value is used.

`-Q "[exit_code ...] [EXCLUDE(exit_code ...)]"`

Specify automatic job requeue exit values. Use spaces to separate multiple exit codes. The reserved keyword `all` specifies all exit codes. Exit codes are typically between 0 and 255. Use a tilde (`~`) to exclude specified number or numbers from the list.

`exit_code` has the following form:

```
"[all] [~number ...] | [number ...]"
```

Job level exit values override application-level and queue-level values.

Jobs running with the specified exit code share the same application and queue with other jobs.

Define an exit code as `EXCLUDE(exit_code)` to enable exclusive job requeue. Exclusive job requeue does not work for parallel jobs.

If `mbatchd` is restarted, it does not remember the previous hosts from which the job exited with an exclusive requeue exit code. In this situation, it is possible for a job to

be dispatched to hosts on which the job has previously exited with an exclusive exit code.

lshosts

- `ncpus` displays the number of processors on the host. If `LSF_ENABLE_DUALCORE=Y` in `lsf.conf` for multi-core CPU hosts, displays the number of cores instead of physical CPUs. If EGO is enabled in the LSF cluster and `EGO_DEFINE_NCPUS` is specified in `lsf.conf` or `ego.conf`, the appropriate value for `ncpus` is displayed, depending on the value of `EGO_DEFINE_NCPUS`:
 - `EGO_DEFINE_NCPUS=procs`—`ncpus`=number of processors
 - `EGO_DEFINE_NCPUS=cores`—`ncpus`=number of processors x number of cores per processor
 - `EGO_DEFINE_NCPUS=threads`—`ncpus`=number of processors x number of cores per processor x number of threads per core

`EGO_DEFINE_NCPUS=cores` is the same as setting `LSF_ENABLE_DUALCORE=Y`.
- `nprocs` displays the number of physical processors per CPU configured on a host.
- `ncores` displays the number of cores per processor configured on a host.
- `nthreads` displays the number of threads per core configured on a host.

lsinfo

- If `CONSUMABLE` is Yes the resource is a static or dynamic numeric resource that is specified as consumable in the Resource section of `lsf.shared`.
- `lsinfo` lists the following new resources available in your cluster:
 - `ncpus`—Number of CPUs
 - `nprocs`—Number of physical processors
 - `ncores`—Number of cores per physical processor
 - `nthreads`—Number of threads per processor

lsplace

`-R res_req` displays hosts with the specified resource requirements. When `LSF_STRICT_RESREQ=Y` in `lsf.conf`, LSF rejects resource requirement strings where an usage section contains a non-consumable resource.

New files

No new files have been added in Platform LSF Version 7 Update 3.

New and changed accounting and job event fields

lsb.events

The following records are new in `lsb.events`:

- `GRP_ADD`—Created when a job group is added
- `GRP_MOD`—Created when a job group is modified

The following fields are new or changed in the `JOB_NEW` and the `JOB_MODIFY2` records:

postExecCmd (%s)

Post-execution command to run on the execution host after the job finishes

runtimeEstimation (%d)

Estimated run time for the job

requeueEValues (%s)

Job exit values for automatic job requeue

Bugs fixed since November 2007

Bugs fixed in the May 2008 update (LSF 7 Update 3) since the November 2007 update (LSF 7 Update 2) are listed in the document *Fixed Bugs for Platform LSF 7 Update 3*.

Known Issues

Platform LSF Version 7 Update 3

JOBS limit and hostgroups

Because the host group limit counter is not necessarily equal to the sum of the limits of its members, the JOBS limit only counts the host group that the first execution host belongs to. If a job spans multiple host groups, the host groups do not count towards the limit except of the host group of the first execution host.

For example, for host groups like the following:

- hgrp1: hostA has the JOBS limit = 1
- hgrp2: hostB has the JOBS limit = 1
- hgrp3: hostC has the JOBS limit = 1

and job submission:

```
bsub -n3 -m "hostA! others" -R"span[ptile=1]" sleep 1000
```

`blimits` only shows hgrp1 is 1/1, and limits on hgrp2 and hgrp3 are not triggered.

After the following job is submitted:

```
bsub -n 2 -m hgrp1 sleep 100
```

`blimits` shows that hostA has triggered a limit on hgrp1, and hostB will be ignored for accounting on hgrp1:

```
HOSTS JOBS
hgrp1 1/1
h1     1/1
h2     1/1
```

PMC performance slow in Internet Explorer 7

Internet Explorer does not properly release JavaScript and HTML DOM object memory. Logging on and logging off PMC in the same browser session can cause the browser to hang or respond slowly.

To avoid the problem, close the Internet Explorer browser when logging off PMC, or upgrade to Internet Explorer 8.

Supported platforms for PMC

The PMC can only run on the following platforms: LINUX86, X86_64, NTX86, NTX64. If your LSF cluster includes other platforms, the administrator must manually modify `eservice/esc/conf/service/gui_service.xml` by changing:

```
<ego: ResourceRequirement>select (!NTIA64) </ego: ResourceRequirement>
```

to

```
<ego: ResourceRequirement>select ('LINUX86' || 'X86_64' || 'NTX86' || 'NTX64') </ego: ResourceRequirement>
```

You must restart EGO to make the change take effect.

LSB_LOCALDIR for duplicate event logging must exist

If `LSB_LOCALDIR` in `lsf.conf` specifies a directory that does not exist, `mbatchd` dies after 45 minutes.

Make sure that `LSB_LOCALDIR` specifies a valid path to a local directory that exists only on the first LSF master host (the first host configured in `lsf.cluster.cluster_name`).

Platform LSF Session Scheduler

Job warning action

When using the `bsub -wa` option with the Session Scheduler, you should use `-wa 'INT'`. *Do not* use `-wa 'KILL'`. The 'INT' signal ensures that `ssched` can properly terminate all tasks and clean up all temporary files before exiting.

Documentation errata

EGO_DEFINE_NCPUS

`EGO_DEFINE_NCPUS=procs` is the correct default in `lsf.conf`. The *Platform LSF Configuration Reference* incorrectly shows the default value as `EGO_DEFINE_NCPUS=cores`.

Download the Platform LSF Version 7 Distribution Packages

Download the LSF distribution packages two ways:

- Through FTP at `ftp.platform.com`
- Through the World Wide Web at `my.platform.com`

Download LSF through FTP

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

1. Log on to the LSF file server.
2. Change to the directory where you want to download the LSF distribution files. Make sure that you have write access to the directory. For example:

```
# cd /usr/share/lsf/tarfiles
```

- FTP to the Platform FTP site:

```
# ftp ftp.platform.com
```

- Provide the login user ID and password provided by Platform.
- Change to the directory for the LSF Version 7 release:

```
ftp> cd /distrib/7.0
```

- Set file transfer mode to binary:

```
ftp> binary
```

- For LSF on UNIX and Linux, get the installation distribution file.

```
ftp> get platform_lsf_update3/lsf7update3_lsfinstall.tar.Z
```

Tip:

Before installing LSF on your UNIX and Linux hosts, you must uncompress and extract lsf7update3_lsfinstall.tar.Z to the same directory where you download the LSF product distribution tar files.

- Get the distribution packages for the products you want to install on the supported platforms you need. For example:

- For the Solaris 7 64-bit version of LSF Version 7:

```
ftp> get platform_lsf_update3/lsf7update3_sparc-sol7-64.tar.Z
```

Tip:

Put the LSF distribution files in the same directory as the installation tar files. *Do not* uncompress and extract the distribution files.

- For 32-bit LSF Version 7 on Windows:

```
ftp> get platform_lsf_update3/lsf7update3_win32.msi
```

- Download the Platform LSF Version 7 documentation from /distrib/7.0/docs/.

```
ftp> get docs/lsf7update3_documentation.zip
```

```
ftp> get docs/lsf7update3_documentation.tar.Z
```

Tip:

After installing LSF, you should extract the Platform LSF Version 7 documentation files to LSF_TOP/docs/lsf. Browse LSF_TOP/docs/lsf/index.html to access the LSF 7 Knowledge Center. If you install the Platform Management Console, the LSF 7 Knowledge Center is installed automatically to LSF_TOP/docs/lsf.

- Download the Platform EGO Version 1.2.3 documentation from /distrib/7.0/docs/.

```
ftp> get docs/ego1.2.3_documentation.zip
```

```
ftp> get docs/ego1.2.3_documentation.tar.Z
```

Tip:

After installing LSF, you should extract the EGO documentation files to LSF_TOP/docs/ego. Browse LSF_TOP/docs/ego/index.html to access the EGO Knowledge Center. If you install the Platform Management Console, the EGO

Knowledge Center is installed automatically to LSF_TOP/docs/ego.

11. Optional. Download the Platform Management Console (PMC) distribution package from `/distribution/7.0/platform_lsf_update3/`.

```
ftp> get platform_lsf_update3/lsf7Update3_pmc_linux-x86.tar.Z
```

OR

```
ftp> get platform_lsf_update3/lsf7Update3_pmc_linux-x86_64.tar.Z
```

Note:

To take advantage of the Platform LSF reporting feature, you *must* download and install the Platform Management Console. The reporting feature is only supported on the same platforms as the Platform Management Console: 32-bit and 64-bit x86 Windows and Linux operating systems.

12. Exit FTP.

```
ftp> quit
```

Download LSF from my.platform.com

You must provide your Customer Support Number and register a user name and password on `my.platform.com` to download LSF.

To register at `my.platform.com`, click `New User?` and complete the registration form. If you do not know your Customer Support Number or cannot log in to `my.platform.com`, send email to `support@platform.com`.

1. Navigate to `http://my.platform.com`.
2. Choose `Products > Platform LSF Family > LSF 7 Update 3`.
3. Under `Download`, choose `Product Packages`.
4. Select the `Updates, Packages, and Documentation` you wish to download.
5. Log out of `my.platform.com`.

Archive location of previous update releases

Directories containing release notes and distribution files for previous LSF Version 7 update releases are located on the Platform FTP site under `/distribution/7.0/archive`. Archive directories are named relative to the current update release:

- LSF Version 7 Update 1: `/distribution/7.0/archive/update1`
- LSF Version 7 Update 2: `/distribution/7.0/archive/update2`

Install Platform LSF Version 7

Installing Platform LSF involves the following steps:

1. Get a DEMO license (`license.dat` file).
2. Run the installation programs.

Get a Platform LSF demo license

Before installing Platform LSF Version 7, you must get a demo license key.

Contact `license@platform.com` to get a demo license.

Put the demo license file `license.dat` in the same directory where you downloaded the Platform LSF product distribution tar files.

Run the UNIX and Linux installation

Use the `lsfinstall` installation program to install a new LSF Version 7 cluster, or upgrade from an earlier LSF version.

See *Installing Platform LSF on UNIX and Linux* for new cluster installation steps.

See the *Platform LSF Command Reference* for detailed information about `lsfinstall` and its options.

Important:

DO NOT use the UNIX and Linux upgrade steps to migrate an existing LSF 7 cluster or LSF 7 Update 1 cluster to LSF 7 Update 3. Follow the manual steps in the document *Migrating to Platform LSF Version 7 Update 3 on UNIX and Linux* to migrate an existing LSF 7 Update 1 cluster to LSF 7 Update 3 on UNIX and Linux.

Run the Windows installation

Platform LSF on Windows 2000, Windows 2003, and Windows XP is distributed in the following packages:

- `lsf7update3_win32.msi`
- `lsf7update3_win-x64.msi`
- `lsf7update3_win-ia64.msi`

See *Installing Platform LSF on Windows* for new cluster installation steps.

To migrate your existing LSF Version 7 cluster on Windows to LSF 7 Update 3, you must follow the manual steps in the document *Migrating Platform LSF Version 7 to Update 3 on Windows* (`lsf_migrate_windows_to_update3.pdf`).

Install Platform LSF License Scheduler

See *Using Platform LSF License Scheduler* for installation and configuration steps.

Install Platform LSF Session Scheduler

See *Installing and Running Platform LSF Session Scheduler* for installation and configuration steps.

Install Platform LSF HPC

Use `lsfinstall` to install a new Platform LSF HPC cluster or to upgrade LSF HPC from a previous release.

Important:

Make sure `ENABLE_HPC_INST=Y` is specified in `install.config` to enable Platform LSF HPC installation and configuration.

See *Using Platform LSF HPC* for installation and configuration steps.

Install Platform LSF Desktop Support

See the *Platform LSF Desktop Support Administrator's Guide* for installation and configuration steps.

Learn About Platform LSF Version 7

Information about Platform LSF is available from the following sources:

- World Wide Web and FTP
- Platform LSF documentation
- Platform EGO documentation
- Platform training

World Wide Web and FTP

Information about Platform LSF Version 7 is available in the LSF area of the Platform FTP site (<ftp.platform.com/distrib/7.0/>).

The latest information about all supported releases of Platform LSF is available on the Platform Web site at www.platform.com.

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

my.platform.com

my.platform.com—Your one-stop-shop for information, forums, e-support, documentation and release information. my.platform.com provides a single source of information and access to new products and releases from Platform Computing.

On the Platform LSF Family product page of my.platform.com, you can download software, patches, updates and documentation. See what's new in Platform LSF Version 7, check the system requirements for Platform LSF, or browse and search the latest documentation updates through the Platform LSF Knowledge Center.

Platform LSF documentation

The Platform LSF Knowledge Center is your entry point for all LSF documentation. If you have installed the Platform Management Console, access and search the Platform LSF documentation through the link to the Platform Knowledge Center.

Get the latest LSF documentation from my.platform.com. Extract the LSF documentation distribution file to the directory `LSF_TOP/docs/lsf`.

Platform EGO documentation

The Platform EGO Knowledge Center is your entry point for Platform EGO documentation. It is installed when you install LSF. To access and search the EGO documentation, browse the file `LSF_TOP/docs/ego/1.2.3/index.html`.

If you have installed the Platform Management Console, access the EGO documentation through the link to the Platform Knowledge Center.

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 at www.platform.com/Services/Training/, or contact Training@platform.com for details.

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 Inc. 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/company/contact-us for other contact information.

Get patch updates and other notifications

To get periodic patch update information, critical bug notification, and general support notification from Platform Support, contact supportnotification-request@platform.com with the subject line containing the word "subscribe".

To get security related issue notification from Platform Support, contact securenotification-request@platform.com with the subject line containing the word "subscribe".

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 Inc. 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)

Copyright

© 1994-2008, Platform Computing Inc.

Although the information in this document has been carefully reviewed, Platform Computing Inc. (“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 policy

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.

POWERING HIGH PERFORMANCE, PLATFORM COMPUTING, PLATFORM SYMPHONY, PLATFORM JOBSCHEDULER, 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, and FLEXlm are registered trademarks or trademarks of Macrovision Corporation in the United States of America and/or other countries.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

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/legal-notices/third-party-license-agreements