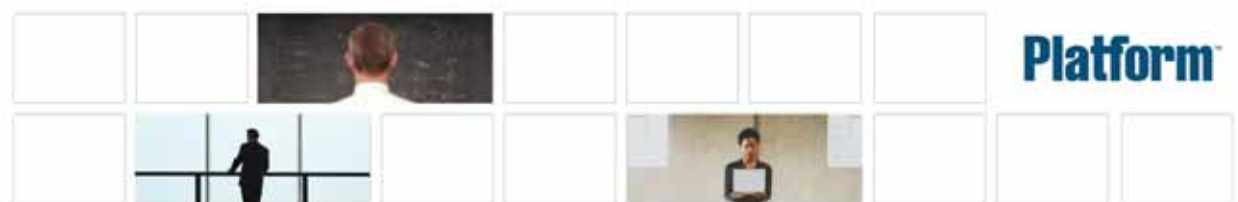

Release Notes for Platform™ LSF™ Version 7 Update 1

Release date: June 2007
Last modified: January 31, 2008
Comments to: doc@platform.com
Support: support@platform.com

Contents

- ◆ [What's New in Platform LSF Version 7 Update 1—June 2007](#) on page 2
- ◆ [Upgrade and Compatibility Notes](#) on page 6
- ◆ [What's Changed in Platform LSF Version 7 Update 1](#) on page 10
- ◆ [Known Issues](#) on page 33
- ◆ [Download the Platform LSF Version 7 Distribution Packages](#) on page 35
- ◆ [Install Platform LSF Version 7](#) on page 38
- ◆ [Learn About Platform LSF Version 7](#) on page 40
- ◆ [Get Technical Support](#) on page 42



What's New in Platform LSF Version 7 Update 1—June 2007

- ◆ Performance, scalability, reliability, usability enhancements
- ◆ LSF on Platform EGO
- ◆ Scheduling enhancements
- ◆ LSF reports built on EGO
- ◆ LSF License Scheduler
- ◆ Platform LSF Desktop Support

For more information

For detailed information about what's new in Platform LSF Version 7 Update 1, visit the Platform Computing Web site to see *Features, Benefits & What's New*.

Performance, scalability, reliability, usability enhancements

- ◆ *Improved performance monitoring*—Collect and display real-time performance metrics with new `badmin perfmon` command. Configure monitoring to sample metric data as frequently as you need, and log data to a file for later analysis. Display performance of job submission requests and information queries for jobs, hosts, and queues.
- ◆ *Patch installation facility*—For improved cluster maintainability and version management. Use the new LSF installer to update your existing LSF Version 7 cluster to LSF Version 7 Update 1 on UNIX and Linux. When you run `lsfinstall`, the new LSF installer calls the new patch installer, which includes functionality to manage updates to a licensed Platform cluster. The LSF installer also makes any cluster configuration updates that are necessary. Later, you can use the patch install command `patchinstall` to check contents of a package and compatibility with the cluster, and patch or roll back a cluster. The version command `pversions` is also provided to query the patch history and show information about cluster and product version and patch levels. Use `pversions` to query a cluster or check contents of a package. How to use the patch installation facility is described in “Cluster Version Management and Patching on UNIX and Linux” in *Administering Platform LSF*. See [Update your existing LSF Version 7 cluster to Version 7 Update 1 on UNIX and Linux](#) on page 7 for more information.
- ◆ *Improved control of job exit handling*—Optionally *exclude* jobs exited because of user actions or LSF-related policies from the job exit calculation, include all job exits *except* those that are related to user action or LSF policy, or include *all* job exit cases in the exit rate count. Set global exit rate for all hosts in the cluster, and configure exit rate for a host based on the number of slots.
- ◆ `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. Register at `my.platform.com`, and login to download software, patches, updates and documentation. See what's new in Platform LSF, check the system requirements for Platform LSF, and browse the latest documentation updates through the Platform LSF Knowledge Center.

LSF on Platform EGO

- ◆ *LIM enhancements*—The LIM daemon automatically collects information about hosts in an EGO cluster, and accurately determines running host models and types. Platform EGO version 1.2.2 increases the number of host types you can manually define in `EGO_CONFDIR/ego.shared` from 128 to 1024. If `ego.shared` is not fully defined with all known host models and types found in the EGO cluster, LIM attempts to match an unrecognized running host to one of the models and types that is defined. LIM supports both exact matching of host models and types, and "fuzzy" matching (where an entered host model name or type is slightly different from what is defined in `ego.conf`).
- ◆ *Processor, core, and thread detection*—If `EGO_DEFINE_NCPUS` is specified in `ego.conf`, `lshosts` displays the appropriate value for `ncpus`, depending on the value of `EGO_DEFINE_NCPUS`:
 - ❖ When `EGO_DEFINE_NCPUS=procs`, the value of `ncpus`=number of processors
 - ❖ When `EGO_DEFINE_NCPUS=cores`, the value of `ncpus`=number of processors * the number of cores
 - ❖ When `EGO_DEFINE_NCPUS=threads`, the value of `ncpus`=number of processors * number of cores * number of threads

Scheduling enhancements

- ◆ *Absolute job priority scheduling (APS)*—Control job dispatch order to prevent job starvation. APS requires a Platform_HPC license. When configured in a queue, APS sorts pending jobs for dispatch according to a job priority value calculated based on several configurable job-related factors in a single priority formula. Each job priority weighting factor configured in the queue can contain subfactors. Factors and subfactors can be independently assigned a weight. APS provides administrators with detailed yet straightforward control of the job selection process:
 - ❖ Every job in an APS queue has a dynamically calculated priority value.
 - ❖ Job priority is calculated for pending jobs across multiple queues based on the sum of configured factor and subfactor values. Jobs are then ordered based on the calculated APS value.
 - ❖ APS only sorts the jobs according to the calculated priority; job scheduling is still based on configured LSF scheduling policies. LSF attempts to schedule and dispatch jobs based on their order in the APS queue.
 - ❖ You can adjust the following APS factors:
 - ❖ A weight for scaling each job related factor and subfactor
 - ❖ Limits for some job related factors
 - ❖ A grace period for each factor and subfactor
 - ❖ Administrators can configure absolute priority scheduling (APS) across multiple queues in APS queue groups. A queue group uses a single formula to calculate APS values. Jobs submitted to queues in the queue group are dispatched based on the priority of the master queue in the group.

- ❖ Administrators can also set a static system APS value for a job. A job with a system APS priority is guaranteed to have a higher priority than any calculated value. Jobs with higher system APS settings have priority over jobs with lower system APS settings.
- ❖ Administrators can use the ADMIN factor to manually adjust the calculated APS value for individual jobs.
- ◆ *blaunch distributed application framework*—Most MPI implementations and many distributed applications use `rsh` and `ssh` as their task launching mechanism. The `blaunch` command provides a drop-in replacement for `rsh` and `ssh` as a simplified and transparent method for launching parallel and distributed applications within LSF. `blaunch` transparently connects directly to the RES/SBD on the remote host, and subsequently creates and tracks the remote tasks, and provides the connection back to LSF. There no need to insert `pam`, `taskstarter` or any other wrapper. `blaunch` only works under LSF. It can *only* be used to launch tasks on remote hosts that are part of a job allocation. It *cannot* be used as a standalone command. `blaunch` is not supported on Windows.
- ◆ *Backfill window scheduling*—Use the new `bslots` command to see unused capacity by displaying slots available for parallel jobs and advance reservations, and the available run time for those slots. The available slots are not currently used for running jobs, and provide a convenient window for submitting backfill jobs. The calculation of the backfill window is based on information about the current running jobs, slot reservations, advance reservations obtained from `mbatchd`.
- ◆ *Support for multiple resource requirement strings (-R) options*—Enables administrators to more easily change and add resource requirements and to simplify the use of scripts for job submission. You can now specify multiple `-R` resource requirement strings on `order`, `same`, `usage`, and `select` sections. You can specify multiple strings instead of using the `&&` operator. LSF merges the multiple `-R` options into one string and selects a host that meets all of the resource requirements. The number of `-R` option sections is unlimited, up to a maximum of 512 characters for the entire string.

LSF reports built on EGO

The LSF reporting feature adds the following new data loader plugins for LSF desktop support:

- | | |
|-----------------------------|---|
| Desktop job | The desktop job data loader (<code>desktopjobdataloader</code>) is a polling loader that loads job completion logs from each desktop server and loads this data into the <code>ACTIVE_DESKTOP_JOBDATA</code> table. This data loader is only available on Linux hosts. By default, this data loader loads data every day. |
| Desktop client | The desktop client data loader (<code>desktopclientdataloader</code>) is a polling loader that samples client status data from the <code>WSCliStatus</code> file and loads this data into the <code>ACTIVE_DESKTOP_SED_CLIENT</code> table. This data loader is only available on Linux hosts. By default, this data loader samples data every ten minutes. |
| Desktop active event | The desktop active event data loader (<code>desktopeventloader</code>) is a polling loader that collects data on downloaded and reported jobs from the desktop <code>event.log</code> files. For each event of type 2 (<code>REPORT_JOB</code>) and type 4 (<code>COMPLETE_JOB</code>), |

`desktopeventloader` loads this data into the `ACTIVE_DESKTOP_ACEVENT` table. This data loader is only available on Linux hosts. This data loader collects data when an event is logged into the `event.log` files.

LSF License Scheduler

- ◆ Use *license feature locality* to limit features from different service domains to a specific cluster, so that License Scheduler does not grant tokens to jobs from licenses that cannot be used on the cluster requesting the token.

Platform LSF Desktop Support

- ◆ *Improved SED debugging and job diagnosis*—Selected SED log messages are written into a new category `SED` in the Windows Events Log system. Debug mode can now be enabled at job level for individual SEDs. LSF Desktop now supports `bsub -o` and `bsub -e`.
- ◆ *Simplified service management and logging*—Enable Platform EGO management of the Platform LSF desktop support master execution daemon (MED) and the Web server components (Tomcat and Apache) so that they run as EGO services. This enables automatic startup of these services, which improves availability.
- ◆ *Improved job visibility and more consistent information display*—LSF Desktop Web Server (WS) is extended to report current status of all active SEDs, including idle, running, suspended, stopped jobs, and opt out. Accurate statistics about Desktop jobs are displayed on the `Host statistics today information` page. WS reports additional status items: number of jobs downloaded from the MBD, number of jobs dispatched and redispached.
- ◆ *MED/Web Service high availability*—provides a way to maximize CPU usage by ensuring that successfully completed rerunnable jobs run only once, even if master execution daemon (MED) and Web server (WS) processes fail during job execution. With high availability enabled, client hosts can upload job results to a new desktop server if they can no longer connect to the original desktop server.

Upgrade and Compatibility Notes

- ◆ Server host compatibility Platform LSF
- ◆ Upgrade from an earlier version of LSF on UNIX and Linux
- ◆ Update your existing LSF Version 7 cluster to Version 7 Update 1 on UNIX and Linux
- ◆ Migrate LSF on Windows
- ◆ Maintenance pack and update availability
- ◆ System requirements
- ◆ API compatibility
- ◆ Multiple cluster configuration
- ◆ NCPUS detection on AIX
- ◆ Enable the full Platform Management Console

Server host compatibility Platform LSF

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

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

Add new IBM AIX host types and models

Platform LSF Version 7 Update 1 supports a new host type and model for IBM AIX 5.3 POWER6 hosts. For LIM to correctly identify the new host type and model, you must manually add them to `lsf.shared`.

-
- 1 Edit `lsf.shared` and add the new host type IBM9117 in the HostType section.
Begin HostType
IBM9117
End HostType
 - 2 Edit `lsf.shared` and add new host model PowerPC_Power6 in the HostModel section.
Begin HostModel
PowerPC_Power6 14.0 (IBM9117)
End HostModel
 - 3 Restart the master lim and slave lims running on all hosts to pick up the new host type and model.
-

Upgrade from an earlier version of LSF on UNIX and Linux

Run `lsfinstall` to *upgrade* to LSF Version 7 from an earlier version of LSF on UNIX and Linux. Follow the steps in *Upgrading Platform LSF on UNIX and Linux*.

IMPORTANT: Do not use the UNIX and Linux upgrade steps to update an existing LSF Version 7 cluster to LSF Version 7 Update 1. Follow the steps the “Cluster Version Management and Patching on UNIX and Linux” chapter in *Administering Platform LSF* to update your existing LSF Version 7 cluster to LSF Version 7 Update 1.

Update your existing LSF Version 7 cluster to Version 7 Update 1 on UNIX and Linux

You must use the latest `lsfinstall` program and the latest `install.config` template file to *update* your cluster. Follow the steps in the “Cluster Version Management and Patching on UNIX and Linux” chapter in *Administering Platform LSF* to update your existing LSF Version 7 cluster to LSF Version 7 Update 1.

IMPORTANT: Before running `lsfinstall`, you must download and extract the new installation distribution file for LSF Version 7 Update 1: `lsf7.0.1_lsfinstall.tar.Z` to use the latest version of `lsfinstall`. Prepare the `install.config` file using the new template and information from your original installation. The new template has additional parameters for the LSF Version 7 patch installation and management facility.

Migrate LSF on Windows

To migrate an LSF on Windows to LSF Version 7 from an earlier version of LSF on Windows, follow the steps in “Migrate Your Windows Cluster to Platform LSF Version 7” (`lsf_migrate_windows.pdf`).

Maintenance pack and update availability

At release, Platform LSF Version 7 Update 1 includes all bug fixes and solutions up to and including bug fixes and solutions before June, 2007. Fixes after June 2007 will be included in the next LSF update.

Fixes in the November 2006 Maintenance Pack were included in the March 2007 update.

As of February 2007, monthly maintenance packs are no longer distributed for LSF Version 7.

System requirements

See the Platform Computing Web site for information about supported operating systems and system requirements for the Platform LSF family of products:

- ◆ Platform LSF
- ◆ Platform LSF License Scheduler

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, including job submission using JSDL and IPv6 address formats, 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 following new APIs have been added for LSF Version 7 Update 1:

- ◆ `lsb_getalloc()`
- ◆ `lsb_launch()`
- ◆ `ls_getmyhostname2()`

The following APIs have changed for LSF Version 7 Update 1:

- ◆ `lsb_modify()` and `lsb_submit()`—add field `apsString` to the `submit` structure
- ◆ `lsb_queueinfo()`—adds fields to the `queueInfoEnt` structure:
 - ❖ `queueGroup`
 - ❖ `numApsFactors`
 - ❖ `apsFactorInfoList`
 - ❖ `apsFactorMaps`
 - ❖ `apsLongNames``lsb_queueinfo()` also adds the `Q_ATTRIB_APS` value to the `qAttrib` field
- ◆ `lsb_readjobinfo()` and `lsb_readjobinfo_cond()`—add fields to the `jobInfoEnt` structure:
 - ❖ `aps`
 - ❖ `adminAps`
 - ❖ `adminFactorVal`

Multiple cluster configuration

In Platform LSF Version 7, multiple independent clusters can no longer share the same configuration directory. You must install each LSF cluster in a unique location.

NCPUS detection on AIX

On a machine running AIX, `ncpus` detection is different from previous release. Under AIX, the number of detected physical processors is always 1, whereas the number of detected cores is the number of cores across all physical processors. Thread detection is the same as other operating systems (the number of threads per core).

Enable the full Platform Management Console

By default, only the LSF reporting feature is enabled in the Platform Management Console (PMC) after installation. Complete the following steps to enable the full PMC functionality.

-
- 1 With an XML editor, open `pmc_conf_ego.xml`.
 - ❖ **Windows:** `EG0_TOP\gui\conf\pmcconf\pmc_conf_ego.xml`
 - ❖ **Linux:** `EG0_TOP/gui/conf/pmcconf/pmc_conf_ego.xml`

If you ran `egoconfig mghost`, then `pmc_conf_ego.xml` is located in the `EG0share` directory:

 - ❖ **Windows:** `EG0share\gui\conf\pmcconf\pmc_conf_ego.xml`
 - ❖ **Linux:** `EG0share/gui/conf/pmcconf/pmc_conf_ego.xml`
 - 2 In the configuration section, locate the parameter:
`<Name>OnlyShowReport</Name>`.
 - 3 In the `<Name>` parameter, change `<Value>true</Value>` to `<Value>>false</Value>`.
 - 4 Save and close `pmc_conf_ego.xml`.
 - 5 Restart the WEBGUI service.
-

What's Changed in Platform LSF Version 7 Update 1

- ◆ Changed behavior
- ◆ New and changed configuration parameters and environment variables
- ◆ New and changed commands, options, and output
- ◆ New and changed files
- ◆ New and changed accounting and job event fields
- ◆ LSF daemon management
- ◆ Directory structure changes
- ◆ Bugs fixed since March 2007

Changed behavior

Banded licensing

The memory limit for S-Class licenses on X86/AMD64/EM64T processors has increased from 8 GB to 16 GB. The other classes of licenses have not changed.

You can use permanent licenses with restrictions in operating system and hardware configurations. These banded licenses have three classes, with the E-class licenses having no restrictions.

Banded licenses now support the following operating systems and hardware configurations:

License type	Supported operating systems	Processor	Physical memory	Physical processors/sockets
B-Class	Linux, Windows, MacOS	Intel X86/AMD64/EM64T	Up to and including 4 GB physical memory on a node	Up to and including 2 processors
S-Class	Linux, Windows, MacOS	Intel X86/AMD64/EM64T	Up to and including 16 GB physical memory on a node	Up to and including 4 processors
E-Class	Linux, Windows, MacOS	Intel X86/AMD64/EM64T	More than 16 GB physical memory on a node	More than 4 processors
	All other LSF-supported operating systems	Intel X86/AMD64/EM64T	N/A	N/A
	N/A	All other supported processors	N/A	N/A

In the LSF license file:

```
FEATURE lsf_manager lsf_ld 6.200 8-may-2008 2 ADE2C12C1A81E5E8F29C \
VENDOR_STRING=Platform NOTICE=Class(S)
FEATURE lsf_manager lsf_ld 6.200 8-may-2008 10 1DC2C1CCEF193E42B6DC \
VENDOR_STRING=Platform NOTICE=Class(E)
```

Enforcement of dual-core processor licenses on Linux

Dual-core processor hosts running Linux must be licensed by the `lsf_dualcore_x86` license feature.

Each dual core processor requires one standard LSF license and one `lsf_dualcore_x86` license.

Use `lshosts -l` to see the number of dual-core licenses enabled and needed. For example:

```
lshosts -l hostB
HOST_NAME: hostB
type      model  cpuf  ncpus  ndisks  maxmem  maxswp  maxtmp  rexpri  server  nprocs  ncores  nthreads
LINUX86  PC6000  116.1    2      1    2016M  1983M  72917M      0    Yes      1      1      2

...
LICENSES_ENABLED: (LSF_Base LSF_Manager LSF_MultiCluster LSF_Sched_Fairshare
LSF_Sched_Resource_Reservation LSF_Sched_Preemption LSF_Sched_Parallel
LSF_Sched_Advance_Reservation LSF_DualCore_x86)

...
LICENSE CLASS NEEDED: Class(B), Multi-cores
...
```

Enforcement of multicore processor licenses on Linux and Windows

Multicore hosts running Linux or Windows must be licensed by the `lsf_dualcore_x86` license feature. Each physical processor requires one standard LSF license and `num_cores-1` `lsf_dualcore_x86` licenses. For example, a processor with 4 cores requires 3 `lsf_dualcore_x86` licenses.

Use `lshosts -l` to see the number of multicore licenses enabled and needed. For example:

```
lshosts -l hostB
HOST_NAME: hostB
type      model  cpuf  ncpus  ndisks  maxmem  maxswp  maxtmp  rexpri  server  nprocs  ncores  nthreads
LINUX86  PC6000  116.1    2      1    2016M  1983M  72917M      0    Yes      1      1      2

...
LICENSES_ENABLED: (LSF_Base LSF_Manager LSF_MultiCluster LSF_Sched_Fairshare
LSF_Sched_Resource_Reservation LSF_Sched_Preemption LSF_Sched_Parallel
LSF_Sched_Advance_Reservation LSF_DualCore_x86)

...
LICENSE CLASS NEEDED: Class(B), Multi-cores
...
```

Determining what licenses a host needs

Use `lim -t` to see the license requirements for a host. For example:

```
lim -t hostA
Host Type           : NTX64
Host Architecture   : EM64T_1596
Physical Processors  : 2
Cores per Processor : 4
Threads per Core     : 2
License Needed       : Class(B), Multi-cores
Matched Type         : NTX64
Matched Architecture : EM64T_3000
Matched Model        : Intel_EM64T
CPU Factor           : 60.0
```

Resource requirements in application profiles

Job-level, application-level, and queue-level resource requirements are merged in the following manner:

- ◆ If resource requirements are not defined at the application level, job-level and queue-level resource requirements are merged.
- ◆ When application-level resource requirements are defined, job-level requirements usually take precedence. Specifically:
 - ❖ The `select` sections from the job, application profile, and queue must all be satisfied.
 - ❖ The job-level `order` section overrides the application profile section, which overrides the queue-level section.
 - ❖ The job-level `rusage` section takes precedence. Any `rusage` requirements defined in the application profile that are not already specified at the job level are then merged. Any queue-level requirements are then merged with that result.
 - ❖ The job-level `span` section overrides the application profile `span` section, which overrides a queue-level section.
 - ❖ The `same` section from the job, application profile, and queue must all be satisfied.

For internal load indices and duration, jobs are rejected if they specify resource reservation requirements at the job level or application level that exceed the requirements specified in the queue.

If `RES_REQ` is defined at the queue level and there are no load thresholds defined, the pending reasons for each individual load index will not be displayed by `bjobs`.

LSF reporting data loader plug-ins

The LSF reporting feature adds the following new data loader plugins for LSF desktop support:

Desktop job The desktop job data loader (`desktopjobdataloader`) is a polling loader that loads job completion logs from each desktop server and loads this data into the `ACTIVE_DESKTOP_JOBDATA` table. This data loader is only available on Linux hosts. By default, this data loader loads data every day.

Desktop client The desktop client data loader (`desktopclientdataloader`) is a polling loader that samples client status data from the `WSCliantStatus` file and loads this data into the `ACTIVE_DESKTOP_SED_CLIENT` table. This data loader is only available on Linux hosts. By default, this data loader samples data every ten minutes.

Desktop active event The desktop active event data loader (`desktopeventloader`) is a polling loader that collects data on downloaded and reported jobs from the desktop `event.log` files. For each event of type 2 (`REPORT_JOB`) and type 4 (`COMPLETE_JOB`), `desktopeventloader` loads this data into the `ACTIVE_DESKTOP_ACEVENT` table. This data loader is only available on Linux hosts. This data loader collects data when an event is logged into the `event.log` files.

New and changed configuration parameters and environment variables

The following configuration parameters and environment variables are new or changed for LSF Version 7 Update 1:

ego.conf

- ◆ LSF_DAEMONS_CPUS=*lim_cpu_list*—run the EGO LIM daemon on the specified CPUs.
- ◆ EGO_DEFINE_NCPUS—Defines how *ncpus* is computed, and displayed by *lshosts*:

- ❖ When EGO_DEFINE_NCPUS=*procs*, the value of *ncpus*=number of processors
- ❖ When EGO_DEFINE_NCPUS=*cores*, the value of *ncpus*=number of processors * the number of cores
- ❖ When EGO_DEFINE_NCPUS=*threads*, the value of *ncpus*=number of processors * number of cores * number of threads

EGO_DEFINE_NCPUS=*cores* is the same as setting LSF_ENABLE_DUALCORE=Y. LSF_ENABLE_DUALCORE and EGO_ENABLE_DUALCORE are obsolete. Use EGO_DEFINE_NCPUS for improved detection of processors, cores, and threads.

EGO_DEFINE_NCPUS overrides LSF_ENABLE_DUALCORE. If EGO_ENABLE_DUALCORE is set, EGO_DEFINE_NCPUS settings take precedence. When EGO_DEFINE_NCPUS is set, run queue-length values (*r1** values returned by *lslload*) are automatically normalized based on the specified value of EGO_DEFINE_NCPUS. If EGO_DEFINE_NCPUS is not defined, but EGO_ENABLE_DUALCORE is set, the *lim* reports the number of cores. If both EGO_DEFINE_NCPUS and LSF_ENABLE_NCPUS are set, then the EGO_DEFINE_NCPUS takes precedence.

- ◆ EGO_LOCAL_RESOURCES—Override the global definition of how *ncpus* are computed. Use EGO_LOCAL_RESOURCES to override *ncpus* computation on specific dynamic and static hosts. EGO_LOCAL_RESOURCES has the following syntax: EGO_LOCAL_RESOURCES="*[resource resource_name]*". Resource definitions are mutually exclusive. Choose only one resource definition per host. Set *resource_name* to one of the following:
 - ❖ *define_ncpus_procs*
 - ❖ *define_ncpus_cores*
 - ❖ *define_ncpus_threads*

install.config

- ◆ OVERWRITE_PREVIOUS="Y" | "N"—Enables replacement of existing EGO binary files and components (configuration is always preserved, and LSF components are always overwritten). Set the value to "N" if you want to preserve any previously installed EGO binary files or EGO components of the Platform Management Console (PMC) that are found in the current installation directory, instead of replacing them with binaries from the current distribution. This might affect the compatibility or performance of the current software (depending on the version of other Platform products installed). By default, *lsfinstall* overwrites all binaries and components.

- ◆ `PATCH_BACKUP_DIR="/path"`—Full path to the patch backup directory. This parameter is used when you install a new cluster for the first time, and is ignored for all other cases. The file system containing the patch backup directory must have sufficient disk space to back up your files (approximately 400 MB per binary type if you want to be able to install and roll back one enhancement pack and a few additional fixes). It cannot be the root directory (/). If the directory already exists, it must be writeable by the cluster administrator (`lsfadmin`). If you need to change the directory after installation, edit `PATCH_BACKUP_DIR` in `EGO_TOP/patch.conf` and move the saved backup files to the new directory manually. The default patch backup directory is `LSF_TOP/ego/patch/backup`.
- ◆ `PATCH_HISTORY_DIR="/path"`—Full path to the patch history directory. This parameter is used when you install a new cluster for the first time, and is ignored for all other cases. It cannot be the root directory (/). If the directory already exists, it must be writeable by `lsfadmin`. The location is saved as `PATCH_HISTORY_DIR` in `EGO_TOP/patch.conf`. *Do not* change the directory after installation. The default patch history directory is `LSF_TOP/ego/patch`.

lsb.applications

- ◆ `JOB_INCLUDE_POSTPROC=Y | N`—Enables the post-execution processing of the job to be included as part of the job. CPU time and run time of post-execution is included in the job CPU time and run time. `sbatchd` reports the job finish status after post-execution completes, and job finish timestamp is the same time as the job's post-execution completes. That is, `mbatchd` logs the job DONE and the post-execution POST_DONE or POST_ERR status at the same time.
- ◆ `JOB_POSTPROC_TIMEOUT=minutes`—Specifies a timeout in minutes for job post-execution processing. If post-execution processing takes longer than the timeout, `sbatchd` reports the post-execution has failed (POST_ERR status), and kills the process group of the job's post-execution processes. The specified timeout must be greater than zero. Some limitations:
 - ❖ If `JOB_INCLUDE_POSTPROC` is enabled in the application profile, and `sbatchd` kills the post-execution processes because the timeout has been reached, the CPU time of the post-execution processing is set to 0, and the job's CPU time will not include the CPU time of the post-execution processing.
 - ❖ If `JOB_POSTPROC_TIMEOUT` is configured in an application profile, only the parent process of the post-execution command is killed when the timeout expires. The child processes of the post-execution command are not killed.
- ◆ `DJOB_COMMFAIL_ACTION="KILL_TASKS"`—Defines the action LSF should take if it detects a communication failure with one or more remote tasks. If defined, LSF will try to kill all the current tasks of a parallel job associated with the communication failure. If not defined, LSF terminates all tasks and shuts down the entire job. This parameter only applies to the `blaunch` distributed application framework.

- ◆ `DJOB_ENV_SCRIPT=script_name`—Defines the name of a user-defined script for setting and cleaning up the parallel job environment. The specified script must support a setup argument and a cleanup argument. The script will be executed by LSF with the setup argument before launching a parallel job, and with argument cleanup after the parallel job is finished. The script will run as the user, and will be part of the job. If a full path is specified, LSF will use the path name for the execution. Otherwise, LSF will look for the executable from `$LSF_BINDIR`. This parameter only applies to the `blaunch` distributed application framework.
- ◆ `DJOB_HB_INTERVAL=seconds`—Specifies a value in seconds used to calculate the heartbeat interval between the task RES and job RES of a parallel job. This parameter only applies to the `blaunch` distributed application framework. When `DJOB_HB_INTERVAL` is specified, the interval is scaled according to the number of tasks in the job. By default, the interval is equal to `SBD_SLEEP_TIME` in `lsb.params`, where the default value of `SBD_SLEEP_TIME` is 30 seconds.
- ◆ `DJOB_RU_INTERVAL=seconds`—Specifies a value in seconds used to calculate the resource usage update interval for the tasks of a parallel job. This parameter only applies to the `blaunch` distributed application framework. When `DJOB_RU_INTERVAL` is specified, the interval is scaled according to the number of tasks in the job. By default, the interval is equal to `SBD_SLEEP_TIME` in `lsb.params`, where the default value of `SBD_SLEEP_TIME` is 30 seconds.
- ◆ `RTASK_GONE_ACTION="[KILLJOB_TASKDONE | KILLJOB_TASKEXIT] [IGNORE_TASKCRASH]"`—Defines the actions LSF should take if it detects that a remote task of a parallel job is gone. This parameter only applies to the `blaunch` distributed application framework.
 - ❖ `IGNORE_TASKCRASH`—A remote task crashes. LSF does nothing. The job continues to launch the next task.
 - ❖ `KILLJOB_TASKDONE`—A remote task exits with zero value. LSF terminates all tasks in the job.
 - ❖ `KILLJOB_TASKEXIT`—A remote task exits with non-zero value. LSF terminates all tasks in the job.

lsb.modules

- ◆ `schmod_aps`—When configured in `lsb.modules`, `schmod_aps` enables priority scheduling (APS) policies configured by `APS_PRIORITY` in `lsb.queues`. The `schmod_aps` plugin name must be configured after the `schmod_fairshare` plugin name in the PluginModule list, so that the APS value can override the fairshare job ordering decision.

lsb.params

- ◆ `SCHED_METRIC_ENABLE=Y | N`—Enables scheduler performance metric collection. Use `badadmin perfmon stop` and `badadmin perfmon start` to dynamically control performance metric collection. The update is done only if the value for the CPU time, resident memory usage, or virtual memory usage has changed by more than 10 percent from the previous update or if a new process or process group has been created.

- ◆ `SCHED_METRIC_SAMPLE_PERIOD=seconds`—Set a default performance metric sampling period in seconds. Cannot be less than 60 seconds. Use `badmin perfmon setperiod` to dynamically change performance metric sampling period. By default, the sampling period is 60 seconds.

lsb.queues

- ◆ `APS_PRIORITY=WEIGHT[[factor, value [subfactor, value]...]...] LIMIT[[factor, value [subfactor, value]...]...] GRACE_PERIOD[[factor, value [subfactor, value]...]...]`—Specifies calculation factors for absolute priority scheduling (APS). Pending jobs in the queue will be ordered according to the calculated APS value. You must explicitly define a weight for factors or subfactors to include a factor as part of the APS value. If weight of a subfactor is defined, but the weight of parent factor is not defined, the parent factor weight will be set as 1. The `WEIGHT` and `LIMIT` are floating-point values. The default unit of `GRACE_PERIOD` is hours (can also be set to minutes or seconds).
- ◆ `QUEUE_GROUP=queue1, queue2 ...`—Configures absolute priority scheduling (APS) across multiple queues. All queues in a group share an `APS_PRIORITY` definition. Jobs are dispatched according to the APS value for all queues in a queue group. Queue group configuration allows jobs from a lower priority queue to be dispatched before a job in a higher priority queue. One queue serves as the master queue, and the others are slave queues. Only the master queue definition contains the `APS_PRIORITY` configuration. No other queues in the cluster have priority between the highest and lowest priority queues in a queue group. When APS is enabled in the queue with `APS_PRIORITY`, the `FAIRSHARE_QUEUES` parameter is ignored. The `QUEUE_GROUP` parameter replaces `FAIRSHARE_QUEUES`, which is obsolete in LSF 7. `FAIRSHARE_QUEUES` can still be used for non-APS queues.
- ◆ `DEFAULT_EXTSCHED` and `MANDATORY_EXTSCHED`—On SGI Altix ProPack 4 and ProPack 5, you can specify a list of memory node IDs with the `cpuset` external scheduler option "`CPUSET[MEM_LIST=mem_node_list]`". LSF creates a `cpuset` for the job that includes the memory nodes specified by `MEM_LIST` in addition to the local memory attached to the CPUs allocated for the `cpuset`. For example, if "`CPUSET[MEM_LIST=30-40]`", and a 2-CPU parallel job is scheduled to run on CPU 0-1 (physically located on node 0), the job is able to use memory on node 0 and nodes 30-40.

lsf.conf

- ◆ `LSF_DAEMONS_CPUS="mbatchd_cpu_list:mbschd_cpu_list"`—`mbatchd` and `mbschd` run on the specified list of CPUs. An empty list means LSF daemons can run on any CPUs. Use spaces to separate multiple CPUs. By default, `mbatchd` and `mbschd` can run on any CPUs. `mbatchd_cpu_list` defines the list of master host CPUs where the `mbatchd` daemon processes can run (hard CPU affinity). `mbschd_cpu_list` defines the list of master host CPUs where the `mbschd` daemon processes can run. Format the CPU lists as a white-space delimited list of CPU numbers.

- ◆ LSF_DISABLE_LSRUN=y | Y—When defined, RES refuses remote connections from `lsrun` and `lsgun` unless the user is either an LSF administrator or root. For remote execution by root, LSF_ROOT_REX must be defined. Other remote execution commands, such as `ch` and `lsmake` are not affected.
- ◆ LSF_NIOS_MAX_TASKS=*integer*—Specifies the maximum number of NIOS tasks.

lsf.licensescheduler

- ◆ *Parameters section:*
 - ❖ AUTH=Y—enables License Scheduler user authentication for projects for `taskman` jobs.
- ◆ *Feature section:*
 - ❖ LOCAL_TO=*cluster_name* | *location_name*(*cluster_name* [*cluster_name* ...]) —configures token locality for the license feature. You must configure different feature sections for same feature based on their locality. By default, If LOCAL_TO is not defined, the feature is available to all clients and is not restricted by geographical location. When LOCAL_TO is configured, for a feature, License Scheduler treats license features served to different locations as different token names, and distributes the tokens to projects according the distribution and allocation policies for the feature.

lsf.shared

- ◆ A resource name cannot be any of the following reserved names:
`cpu cpuf io logins ls idle maxmem maxswp maxtmp type model status`
`it`
`mem ncpus define_ncpus_cores define_ncpus_procs`
`define_ncpus_threads ndisks pg r15m r15s r1m swap swp tmp ut`

Environment variables

The following environment variables are new in LSF Version 7 Update 1:

- ◆ LSB_DJOB_COMMFALL_ACTION
- ◆ LSB_DJOB_ENV_SCRIPT
- ◆ LSB_RTASK_GONE_ACTION

New and changed commands, options, and output

The following command options and output are new or changed for LSF Version 7 Update 1:

badmin

`perfmon start [sample_period] | stop | view | setperiod sample_period`

Dynamically enables and controls scheduler performance metric collection. Collecting and recording performance metric data may affect the performance of LSF. Smaller sampling periods will result in the `lsb.streams` file growing faster.

The following metrics are collected and recorded in each sample period:

- ◆ The number of queries handled by `mbatchd`
- ◆ The number of queries for each of jobs, queues, and hosts. (`bjobs`, `bqueues`, and `bhosts`, as well as other daemon requests)
- ◆ The number of jobs submitted (divided into job submission requests and jobs actually submitted)
- ◆ The number of jobs dispatched
- ◆ The number of jobs completed
- ◆ The numbers of jobs sent to remote cluster
- ◆ The numbers of jobs accepted by from cluster

bbot

You cannot run `bbot` on jobs pending in an absolute priority scheduling (APS) queue.

bhist

- ◆ If you submitted a job using the `OR (| |)` expression to specify alternative resources, `bhist -l` displays the successful `Execution rusage` string with which the job ran. If you submitted a job with multiple resource requirement strings using the `bsub -R` option for the `order`, `same`, `rusage`, and `select` sections, `bjobs -l` displays a single, merged resource requirement string for those sections, as if they were submitted using a single `-R`.
- ◆ `bhist -l` can display job exit codes. A job with exit code 131 means that the job exceeded a configured resource usage limit and LSF killed the job with signal 3 ($131-128=3$).
- ◆ `bhist -l` can display changes to pending jobs as a result of `bmod -aps`.

bhosts

When `LOCAL_TO` is configured for a license feature in `lsf.licensescheduler`, `bhosts -s` shows different resource information depending on the cluster locality of the features.

bjobs

- ◆ `-aps`—Displays absolute priority scheduling (APS) information for pending jobs in a queue with `APS_PRIORITY` enabled. The APS value is calculated based on the current scheduling cycle, so jobs are not guaranteed to be dispatched in this order. Pending jobs are ordered by APS value. Jobs with system APS values are listed first, from highest to lowest APS value. Jobs with calculated APS values are listed next ordered from high to low value. Finally, jobs not in an APS queue are listed. Jobs with equal APS values are listed in order of submission time. APS values of jobs not in an APS queue are shown with a dash (-).

- ◆ If you submitted a job with multiple resource requirement strings using the `bsub -R` option for the `order`, `same`, `rusage`, and `select` sections, `bjobs -l` displays a single, merged resource requirement string for those sections, as if they were submitted using a single `-R`.
- ◆ If you submitted a job using the `OR (||)` expression to specify alternative resources, `bjobs -l` displays the `Execution rusage` string with which the job runs.
- ◆ For jobs submitted to an absolute priority scheduling (APS) queue, `bjobs -l` shows the `ADMIN` factor value and the system APS value if they have been set by the administrator for the job:

blaunch (new)

Most MPI implementations and many distributed applications use `rsh` and `ssh` as their task launching mechanism. The `blaunch` command provides a drop-in replacement for `rsh` and `ssh` as a transparent method for launching parallel applications within LSF.

`blaunch` supports the following core command line options as `rsh` and `ssh`:

- ◆ `rsh host_name command`
- ◆ `ssh host_name command`

All other `rsh` and `ssh` options are silently ignored.

`blaunch` transparently connects directly to the RES/SBD on the remote host, and subsequently creates and tracks the remote tasks, and provides the connection back to LSF. There no need to insert `pam`, `taskstarter` or any other wrapper.

`blaunch` only works under LSF. It can *only* be used to launch tasks on remote hosts that are part of a job allocation. It *cannot* be used as a standalone command.

`blaunch` is not supported on Windows.

blinfo

When `LOCAL_TO` is configured for a feature in `lsf.licensescheduler`, `blinfo` shows the cluster locality and license token allocation information for the license features.

blstat

When `LOCAL_TO` is configured for a feature in `lsf.licensescheduler`, `blstat` shows the cluster locality information for the license features. For example, with a group distribution configuration `blstat` shows the locality of a license feature configured for various sites.

blusers

When `LOCAL_TO` is configured for a feature in `lsf.licensescheduler`, `blusers` shows cluster locality information for the license features.

bmod

- ◆ Administrators can use `bmod -aps` to adjust or override the APS value for pending jobs. `bmod -apsn` cancels previous `bmod -aps` settings. You cannot combining `bmod -aps` with other `bmod` options.
- ◆ You can now specify multiple `-R` resource requirement strings on `order`, `same`, `rusage`, and `select` sections. The `bmod` command does not support the use of the `||` operator.

bqueues

- ◆ `-l` displays absolute priority scheduling (APS) information for queues configured with `APS_PRIORITY`. Pending jobs in the queue are ordered according to the calculated APS value.

- ◆ `-l` displays queues participating in an absolute priority scheduling (APS) queue group. If both FAIRSHARE and APS_PRIORITY are enabled in the same queue, the FAIRSHARE_QUEUES are not displayed. These queues are instead displayed as QUEUE_GROUP.

bslots (new)

Displays slots available for backfill jobs, and slots reserved for parallel jobs and advance reservations. The available slots are not currently used for running jobs and can be used for backfill jobs. The available slots displayed by `bslots` are only a snapshot of the slots currently not in use by parallel jobs or advance reservations. They are not guaranteed to be available at job submission.

By default, displays all available slots, and the available run time for those slots.

If the available backfill window has no run time limit, its length is displayed as UNLIMITED.

bsub

- ◆ When absolute priority scheduling is configured in the submission queue (APS_PRIORITY in `lsb.queues`), the user-assigned job priority specified by `-sp` is used for the JPRIORITY factor in the APS calculation.
- ◆ You can now specify multiple `-R` resource requirement strings on `order`, `same`, `rusage`, and `select` sections. You can specify multiple strings instead of using the `&&` operator:

```
bsub -R "select[swp > 15]" -R "select[hpx] order[r15m]" -R rusage[mem=100]" -R  
"order[ut]" -R "same[type]" -R rusage[tmp=50:duration=60]" -R "same[model]" myjob
```

LSF merges the multiple `-R` options into one string and selects a host that meets all of the resource requirements. The number of `-R` option sections is unlimited, up to a maximum of 512 characters for the entire string.

- ◆ `-extsched` —On SGI Altix ProPack 4 and ProPack 5, you can specify a list of memory node IDs with the `cpuset` external scheduler option `"CPUSET[MEM_LIST=mem_node_list]"`. LSF creates a `cpuset` for the job that includes the memory nodes specified by `MEM_LIST` in addition to the local memory attached to the CPUs allocated for the `cpuset`. For example, if `"CPUSET[MEM_LIST=30-40]"`, and a 2-CPU parallel job is scheduled to run on CPU 0-1 (physically located on node 0), the job is able to use memory on node 0 and nodes 30-40.

btcp

You cannot run `btcp` on jobs pending in an absolute priority scheduling (APS) queue.

lim

`lim -t` displays host information, such as host type, matched host type, host architecture, physical number of processors, number of cores per physical processor, number of threads per processor core, and license requirements.

NOTE: When running Linux kernel version 2.4, you must run `lim -t` as root to ensure consistent output with other clustered application management commands (for example, output from running `lshosts`).

LIM reads the configuration file `ego.conf` to retrieve configuration information. `ego.conf` is a generic configuration file shared by all daemons and clients. It contains configuration information and other information that dictates the behavior of the software.. `lim` retrieves the following parameters from `ego.conf`:

- ◆ EGO_LIM_PORT—The TCP port `lim` uses to serve all applications.
- ◆ EGO_SERVERDIR—The directory used for reconfiguring `lim`, where the `lim` binary is stored.
- ◆ EGO_LOGDIR—The directory used for message logs.
- ◆ EGO_LOG_MASK—The log level used to determine the amount of detail logged.
- ◆ EGO_DEBUG_LIM—The log class setting for `lim`.
- ◆ EGO_LICENSE_FILE—The full path to and name of the EGO license file
- ◆ EGO_DEFINE_NCPUS—Defines whether `ncpus` is to be defined as `cpus`, `cores`, or `threads`.

lshosts

Host-based default output displays `ncpus`—The number of processors on the host. If `LSF_ENABLE_DUALCORE=Y` in `lsf.conf` for dual-core CPU hosts, displays the number of cores instead of physical CPUs. If `EGO_DEFINE_NCPUS` is specified in `ego.conf`, displays the appropriate value for `ncpus`, depending on the value of `EGO_DEFINE_NCPUS`:

- ◆ When `EGO_DEFINE_NCPUS=procs`, the value of `ncpus`=number of processors
- ◆ When `EGO_DEFINE_NCPUS=cores`, the value of `ncpus`=number of processors * the number of cores
- ◆ When `EGO_DEFINE_NCPUS=threads`, the value of `ncpus`=number of processors * number of cores * number of threads

`EGO_DEFINE_NCPUS=cores` is the same as setting `LSF_ENABLE_DUALCORE=Y`. `LSF_ENABLE_DUALCORE` and `EGO_ENABLE_DUALCORE` are obsolete. Use `EGO_DEFINE_NCPUS` for improved detection of processors, cores, and threads.

Host-based `-l` output displays:

- ◆ `LICENSES_ENABLED`—The licenses that are enabled for each specified host. If `LSF_ENABLE_DUALCORE=Y` in `lsf.conf` for dual-core CPU hosts, `lshosts -l` also displays if dual-core CPU license is enabled for the hosts and the number of dual-core licenses enabled.
- ◆ `LICENSE CLASS NEEDED`—The required banded license class for each specified host. If `LSF_ENABLE_DUALCORE=Y` in `lsf.conf` for dual-core CPU hosts, `lshosts -l` also displays if dual-core CPU license is enabled for the hosts and the number of dual-core licenses needed.
- ◆ If `EGO_DEFINE_NCPUS` is specified in `ego.conf`, displays the appropriate value for `ncpus`, `nprocs`, `ncores`, and `nthreads`.

patchinstall (UNIX—new)

Use to `patchinstall` install and manage patches on an existing licensed Platform cluster. The patch installer includes functionality to query a cluster, check contents of a package and compatibility with the cluster, and patch or roll back a cluster.

For clusters version 7 or earlier, you must obtain the patch installer separately from Platform, and run the `patchinstall` command from your download directory.

For clusters version 7 or later, the patch installer is available under `install` directory under the LSF installation directory. This location may not be in your path, so run the `patchinstall` command from this directory (`LSF_TOP/7.0/install/patchinstall`).

pversions (UNIX—new)

The version command `pversions` is provided to query patch history and deliver information about cluster and product version and patch levels. Use `pversions` to query a cluster or check contents of a package.

By default, `pversions` displays the version and patch level of Platform products. Optionally, the command can also be used to do the following:

- ◆ Check the contents of a package before installing it
- ◆ Show information about a specific Platform product installed
- ◆ Show information about installed packages from specific build
- ◆ Find current versions of a specific Platform file and see information for each

For each binary type, displays basic version information (package build date, build number, package installed date) and lists patches installed (package type, build number, date installed, fixes).

The version command is not located with other LSF commands so it may not be in your path. The command location is `LSF_TOP/7.0/install/pversions`

The cluster location is normally determined by your environment setting, so ensure your environment is set before you run this command (for example, you sourced `profile.platform` or `profile.lsf`).

tspeek

`tspeek` is now supported on Linux hosts. In mixed cluster environments, you can use `tspeek` to monitor job output from a Linux host for a Windows Terminal Services job.

New and changed files

No files have been added or changed in Platform LSF Version 7 Update 1.

New and changed accounting and job event fields

lsb.acct

No fields are new or changed in the `lsb.acct` file records for Platform LSF Version 7 Update 1.

lsb.events

No fields are new or changed in the `lsb.events` file records for Platform LSF Version 7 Update 1.

LSF daemon management

Manage LSF daemons two ways:

- ◆ System management through `rc`, `inittab`, etc.

- ◆ Through Platform EGO Service Controller. If LSF daemons exit unexpectedly, EGO Service Controller automatically restarts and monitors `res` and `sbatchd`.

IMPORTANT: LSF `res` and `sbatchd` do not restart automatically if you run `lsadmin resshutdown` and `badmin hshutdown` to manually shut them down. You must run `lsadmin resstartup` and `badmin hstartup` to restart the daemons after host shutdown.

All LSF commands and tools, including `lsadmin` and `badmin` are available under both management models.

Directory structure changes

The installation directory structure has changed for Platform LSF Version 7. See *Installing Platform LSF on UNIX and Linux* for the details of the new structure. Depending on which products you have installed and platforms you have selected, your directory structure may vary.

Bugs fixed since March 2007

The following bugs have been fixed in the June 2007 update since the March 2007 update:

87551	Date	2007-05-17
	Description	License Scheduler will not count licenses used if the <code>lmsstat -a -c port@host</code> output includes the word "licenses" in it. This can happen if the license server host has the word "licenses" in its domain name or the host checkout license with "licenses" in its name.
	Component	blcollect
	Platform	UNIX
	Impact	Inaccurate license count in License Scheduler
87279	Date	2007-05-14
	Description	Parameter <code>LSF_LICENSE_FILE</code> is not added to <code>lsf.conf</code> file after a new installation
	Component	lsfinstall
	Platform	UNIX
	Impact	Cluster is unlicensed until the <code>LSF_LICENSE_FILE</code> parameter is manually added to the configuration
85528	Date	2007-05-09
	Description	Newer Linux kernels are setting the parent and group of the init process to 1 instead of 0
	Component	All
	Platform	linux
	Impact	This causes problems for pim on Linux since it skips all processes under group 1 including init. <code>bjobs -l</code> does not show correct resource usage. LSF cannot receive resource info from the <code>/tmp/pim.*</code> file since only one process remains.
87100	Date	2007-05-04
	Description	Parallel job using <code>exec rusage</code> pends forever
	Component	<code>schmod_reserve.so</code> <code>schmod_default.so</code> <code>schmod_parallel.so</code>
	Platform	All
	Impact	Job is never dispatched
86438	Date	2007-04-29
	Description	Inconsistent fairshare behavior by restarting and reconfiguring <code>mbatchd</code> after user group is removed
	Component	<code>mbatchd</code>
	Platform	All
	Impact	Fairshare does not work as expected after <code>mbatchd</code> restart

84852	Date	2007-04-29
	Description	Job remains pending forever because of unsatisfied job dependency
	Component	mbatchd
	Platform	All
	Impact	Cannot tell if the job has exited
86348	Date	2007-04-26
	Description	Cannot pass -h and -V as arguments to an MPI program
	Component	pam
	Platform	All
	Impact	Job fails if an MPI program and its options (including -h -V) are not enclosed with single quotes
84289	Date	2007-04-24
	Description	When host exclusion for host partition is defined in queue level, LSF cannot exclude the host defined, and jobs could still be submitted to the excluded host
	Component	mbatchd
	Platform	All
	Impact	In clusters with login hosts and server hosts jobs should not be submitted to run on login hosts. If hosts are licensed to run specialized software, only authorized users should be able to use those hosts. Hosts cannot be excluded hosts from certain queues.
86069	Date	2007-04-23
	Description	epoll_mod error: epoll_ctl() failed. No such file or directory.
	Component	MultiCluster
	Platform	Unix
	Impact	MultiCluster does not work with epoll enabled
84445	Date	2007-04-20
	Description	lsmake hangs or core dumps
	Component	lsmakerm
	Platform	All
	Impact	lsmake fails
82558	Date	2007-04-20
	Description	Slot reservation is not displayed in bjobs, but the job behaves like the reservation is happening
	Component	schmod_cpuset.so

What's Changed in Platform LSF Version 7 Update 1

82558	Date	2007-04-20
	Platform	All
	Impact	Cannot determine the job pending reason easily. It looks like jobs are blocked by certain other jobs. This makes job start time unpredictable.
85618	Date	2007-04-19
	Description	Missing log message
	Component	sbatchd
	Platform	All
	Impact	Hard to tell why job not suspended
70025	Date	2007-04-19
	Description	lsfmon.exe is not in the install package
	Component	Windows installer
	Platform	Windows
	Impact	lsfmon.exe is not in the install package
84904	Date	2007-04-17
	Description	bpeek command fails because it cannot change to the user's home directory
	Component	res
	Platform	All
	Impact	Cannot use bpeek to see the output from the job
84606	Date	2007-04-16
	Description	Job cannot run after being changed with lsb_modify()
	Component	lsb_modify() in libbat.a API
	Platform	All
	Impact	Job cannot run after being changed with lsb_modify()
84908	Date	2007-04-15
	Description	LIM on Linux 2.6 reports wrong pg index
	Component	lim
	Platform	Linux 2.6
	Impact	Wrong load index reported
84745	Date	2007-04-15
	Description	For large remote execution tasks, nios/res gets timeout from time to time
	Component	nios
	Platform	All
	Impact	Remote execution fails

85923	Date	2007-04-13
	Description	Master LIM is very busy after upgrading
	Component	lim
	Platform	All
	Impact	After master LIM is restarted, it takes very long time for all hosts to become ok
72938	Date	2007-04-13
	Description	lsmake fails
	Component	lsmakerm on lsmake
	Platform	UNIX
	Impact	lsmake fails
85689	Date	2007-04-10
	Description	mbschd does not clearly mark the beginning and ending of a scheduling session
	Component	mbschd
	Platform	All
	Impact	Hard to analyze mbschd log file
85323	Date	2007-04-10
	Description	esub cannot change LSB_SUB2_USE_RSV parameter value
	Component	bmod, bsub
	Platform	All
	Impact	Customized esub cannot change user-submitted advanced reservation value
82416	Date	2007-04-09
	Description	bhosts with -l and -s options does not show appropriate column name
	Component	bhosts
	Platform	All
	Impact	Information displayed by bhosts can be misinterpreted
84461	Date	2007-04-04
	Description	mbatchd requests the wrong number of resources from bld when RESOURCE_RESERVE_PER_SLOT is set
	Component	bld, mbatchd
	Platform	All
	Impact	Resources not allocated correctly

What's Changed in Platform LSF Version 7 Update 1

80169	Date	2007-03-27
	Description	The start time recorded in lsb.acct for each job in a chunked set is not the starting time for the job, but rather the time when the chunk of jobs that job was part is sent to the execution host
	Component	mbatchd
	Platform	All
	Impact	lsb.acct file cannot be used to determine the actual wall clock run time of a job
83334	Date	2007-03-23
	Description	bpeek does not work on a Solaris 9 host
	Component	bpeek
	Platform	Solaris 7, 8, 9, 10
	Impact	Cannot get the job output by bpeek on a Solaris 9 host
83520	Date	2007-03-22
	Description	hostsetup does not set LSF startup script correctly
	Component	hostsetup
	Platform	Linux
	Impact	LSF cannot start automatically when the host is rebooting
84207	Date	2007-03-21
	Description	Cannot see the full host name for some hosts
	Component	pam
	Platform	All
	Impact	pam output cannot distinguish between some hosts
83596	Date	2007-03-20
	Description	Cannot remove temp accounting file under LSF_TMPDIR after a job is done
	Component	sbatchd, res
	Platform	All
	Impact	Waste of disk space and file node resources
83986	Date	2007-03-19
	Description	For some "ptile=!" job pending with "Not enough processors to meet the job's spanning requirement", mbdrestart will dispatch them anyway
	Component	mbatchd
	Platform	All
	Impact	Parallel jobs get dispatched unexpectedly

84091	Date	2007-03-13
	Description	The format of JOBID and FACTOR has been changed, causing a display issue in the job exception handling email
	Component	none
	Platform	UNIX/Linux
	Impact	Low
82623	Date	2007-03-13
	Description	mbatchd does not log accurate error message regarding communicating with bld
	Component	bld, mbatchd
	Platform	All
	Impact	Difficult to diagnose the problem
83573	Date	2007-03-07
	Description	When one of the tasks in a POE integrated parallel job exits, the TaskStarter for this task exits, but pam hangs. POE can handle some of the exit codes; e.g. 139 (Segfault), outside of LSF.
	Component	taskstarter
	Platform	All
	Impact	Many applications often have segmentation faults for one or more tasks in a parallel job. LSF should be able to handle this.
83361	Date	2007-03-07
	Description	sbatchd directs load information request to master lim, causing master lim performance penalty
	Component	sbatchd
	Platform	All
	Impact	Master lim becomes slow
83759	Date	2007-03-06
	Description	The event file may be corrupted and job IDs are reused when two mbatchd are running
	Component	mbatchd
	Platform	All
	Impact	More than one job could use the same job ID
83532	Date	2007-03-05
	Description	When LSB_DEFAULTPROJECT environment variable is set, bmod does not work with running jobs
	Component	mbatchd
	Platform	All
	Impact	bmod does not work with running jobs

What's Changed in Platform LSF Version 7 Update 1

83371	Date	2007-03-02
	Description	lsmake will fail in the case of remake with non-zero make level
	Component	lsmake
	Platform	All
	Impact	lsmake fails in the case of remake
83175	Date	2007-03-02
	Description	bsub fails because of XDR error
	Component	bsub
	Platform	All
	Impact	bsub fails
77119	Date	2007-02-27
	Description	mbdrestart changes RUN_TIME in host partition fairshare
	Component	All
	Platform	All
	Impact	User account information or user share priority will be wrong
82742	Date	2007-02-23
	Description	Some CPUs on a host cannot be used after a job got post-done status update without done status update from sbatchd
	Component	All
	Platform	All
	Impact	Some CPUs on a host cannot be used
83221	Date	2007-02-22
	Description	mbatchd core dumps at event replay if one extra JOB_NEW inserted for a job
	Component	All
	Platform	All
	Impact	The cluster is down
80814	Date	2007-02-14
	Description	LIM fails to convert license from lsf_base to lsf_client
	Component	All
	Platform	All
	Impact	Some LSF client hosts are unlicensed
82257	Date	2007-02-13
	Description	lsmake fails with more than 10024 tasks
	Component	lsmake
	Platform	All
	Impact	lsmake is unusable

82254	Date	2007-02-13
	Description	lsmake server (lsmakerm) core dumps, which causes failure of lsmake
	Component	lsmake
	Platform	All
	Impact	lsmake is unusable
80900	Date	2007-02-12
	Description	pam hangs for 15 minutes during shutdown
	Component	pam
	Platform	All
	Impact	pam does exit after the 15 minutes. Having a process hang for this long hurts performance.
80671	Date	2007-02-12
	Description	LSF applies advance reservation defined with -t and current day's index to next week instead of applying it to current day. For example, assuming current time is 6:00 p.m. and today is Wednesday (day 3 in LSF advance reservation day index), the following reservation is defined: brsvadd -m hostA -s -t "3:19:00-3:19:30" LSF will define this reservation to <i>next</i> Wednesday and not today from 7:00 to 7:30.
	Component	All
	Platform	All
	Impact	Advance reservation is not created for the current day
80611	Date	2007-02-12
	Description	After cluster restart, a job will be dispatched by trespassing on an advance reservation on a multi-CPU host
	Component	All
	Platform	All
	Impact	Advance reservation is not honored
77785	Date	2007-02-12
	Description	Events data can be lost if duplicate event logging is enabled together with failover and the primary master becomes unavailable for some time
	Component	All
	Platform	Linux
	Impact	Events data is lost
82345	Date	2007-02-11
	Description	In cross-queue fairshare, CPU time and run time decay too fast
	Component	All

What's Changed in Platform LSF Version 7 Update 1

82345	Date	2007-02-11
	Platform	All
	Impact	Fairshare is not accurate
82770	Date	2007-02-08
	Description	mbschd crashes periodically
	Component	mbschd
	Platform	All
	Impact	Cluster is not operating properly
82739	Date	2007-02-07
	Description	If a large fairshare tree is configured and lots of finished jobs are in lsb.events, mbatchd may take long time to replay
	Component	All
	Platform	All
	Impact	mbatchd does not respond for a long time during mbatchd restart
81748	Date	2007-02-07
	Description	TotalView integration with LAMMPI does not work
	Component	All
	Platform	Linux2.6-glibc2.3-x86_64
	Impact	TotalView integration with LAMMPI does not work

Known Issues

- ◆ Platform LSF Version 7 Update 1
- ◆ Platform LSF on Windows Vista
- ◆ Platform EGO
- ◆ Platform LSF Desktop Support
- ◆ Platform LSF Desktop reporting

Platform LSF Version 7 Update 1

Warning message installing to ACL-enabled file systems

On RHEL5, `lsfinstall` gives a warning message during preinstallation checking if the installation file system has ACL enabled. You should avoid installing LSF on an ACL-enabled file system. To be fixed in a later LSF 7.0 update.

PARALLEL_SCHED_BY_SLOT limitations

- ◆ `PARALLEL_SCHED_BY_SLOT` set in `lsb.params` is not supported for leased-in hosts. Host resources are only updated based on the number of slots on local hosts. Host resources on leased hosts are updated based on the number of CPUs on the leased host, not on the number of slots.
- ◆ `PARALLEL_SCHED_BY_SLOT` set in `lsb.params` causes `JL/P` set in `lsb.users` to be based on slots not CPUs.

Buffer space message on Windows XP

When maximum memory limit is set with the `/3GB` switch in `boot.ini` on Windows XP, some LSF operations (for example query commands like `bqueues` and `bhosts`) give a warning message like:

```
Failed in an LSF library call: Failed in sending/receiving a message:
No buffer space available
```

You should not set the `/3GB` switch in `boot.ini` on LSF master hosts.

Platform LSF on Windows Vista

Cannot delete uninstall directory Windows shows "Access denied" when the local Windows administrator or the cluster administrator tries to delete the LSF uninstall directory. The LSF uninstall directory cannot be deleted because the `C:\LSF_7.0\conf\passwd.lsfuser` file is owned by "System". The `passwd.lsfuser` file must be owned by the cluster administrator.

Shared directory permissions

When users create a shared directory on Windows Vista, the default owner is the directory creator. For LSF to work properly, the shared directory for LSF must be configured so that cluster administrators have read/write permission and all LSF users must have at least read permission. The shared directory must have the following security settings:

- ◆ Owned or co-owned by all cluster administrators
- ◆ Read for all LSF users

cmd.exe permissions For installations on an NTFS file system, users must have "Read" and "Execute" privileges for `cmd.exe`. The following files:

Known Issues

- ◆ %WINDIR%\system32\cmd.exe
- ◆ %WINDIR%\syswow64\cmd.exe

Require the following access permissions:

- ◆ Administrators: Full Control
- ◆ Users: read+execution

Platform EGO

Platform EGO version 1.2.2 increases the number of host types you can be manually define in `EGO_CONFDIR/ego.shared` from 128 to 1024. In a MultiCluster environment where one cluster contains a mix of EGO 1.2.2 hosts and pre-EGO 1.2.2 hosts, the maximum number of host types you can define in `ego.shared` remains 128.

Platform LSF Desktop Support

Platform EGO management of LSF desktop support services applies to the MED and to the Web servers (Tomcat and Apache). With EGO management of LSF desktop support services enabled, you should not use the command `lsfac_daemons` to start or stop Apache or Tomcat services because EGOSC will automatically restart them. Instead, you should use the `egosh` command to start and stop these services.

If EGO management of LSF desktop support services is enabled, you must use an EGO command to start and stop a managed service. From the command line, enter one of the following commands:

```
egosh service start LSFDesktopApache LSFDesktopTomcat
egosh service stop LSFDesktopApache LSFDesktopTomcat
```

Platform LSF Desktop reporting

In the Hourly Desktop Job Throughput report, if an SED host pulls a job from an MED host, but the job failed to run, while another SED host pulls the same job and it runs successfully, the job will be double-counted in both the number of downloaded jobs and the number of completed jobs for the MED host.

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>

IMPORTANT: The latest Platform LSF Version 7 release is Update 2. Distribution packages are available only for Platform LSF Version 7 Update 2 and Platform LSF Version 7 Update 1.

Download LSF through FTP

Prerequisites: 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
```
 - 3 FTP to the Platform FTP site:

```
# ftp ftp.platform.com
```
 - 4 Provide the login user ID and password provided by Platform.
 - 5 Change to the directory for the LSF Version 7 release:

```
ftp> cd /distrib/7.0
```
 - 6 Set file transfer mode to binary:

```
ftp> binary
```
 - 7 For LSF on UNIX and Linux, get the installation distribution file.

```
ftp> get archive/update1/platform_lsf/lsf7.0.1_lsfinstall.tar.Z
```

TIP: Before installing LSF on your UNIX and Linux hosts, you must uncompress and extract `lsf7.0.1_lsfinstall.tar.Z` to the same directory where you download the LSF product distribution tar files.

- 8 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 archive/update1/platform_lsf/lsf7.0.1_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 archive/update1/platform_lsf/lsf7.0.1_win32.msi
```

Download the Platform LSF Version 7 Distribution Packages

- Optional. Download the Platform LSF Version 7 Update 1 documentation.

```
ftp> get archive/update1/docs/lsf7.0.1_documentation.zip
ftp> get archive/update1/docs/lsf7.0.1_documentation.tar.Z
```

NOTE: Get the latest Platform LSF Version 7 documentation from `/distrib/7.0/docs/`.

- Optional. Download the Platform EGO Version 1.2 documentation.

```
ftp> get archive/update1/docs/ego1.2.2_documentation.zip
ftp> get archive/update1/docs/ego1.2.2_documentation.tar.Z
```

NOTE: Get the latest Platform EGO documentation from `/distrib/7.0/docs/`.

- Optional. Download the Platform Management Console (PMC) distribution package from `/distrib/7.0/archive/update1/`.

```
ftp> get
archive/update1/platform_lsf/lsf7.0.1_pmc_linux-x86.tar.Z
OR
ftp> get
archive/update1/platform_lsf/lsf7.0.1_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.

- Exit FTP.

```
ftp> quit
```

Download LSF from my.platform.com

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

If you have not registered 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`.

-
- Navigate to `http://my.platform.com/`.
 - Choose **Products > Platform LSF Family > LSF 7**.
 - Under **Download**, choose **Product Packages**.
 - Select the Updates, Packages, and Documentation you wish to download.
 - 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 `/distrib/7.0/archive`. Archive directories are named relative to the current update release:

- ◆ LSF Version 7 Update 1: `/distrib/7.0/archive/update1`

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, upgrade from and earlier LSF version, or to update your existing LSF Version 7 cluster to LSF Version 7 Update 1.

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.

See the “Cluster Version Management and Patching on UNIX and Linux” chapter in *Administering Platform LSF* for detailed steps for updating your existing LSF Version 7 cluster to LSF Version 7 Update 1.

Run the Windows installation

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

- ◆ `lsf7.0.1_win32.msi`
- ◆ `lsf7.0.1_win-x64.msi`
- ◆ `lsf7.0.1_win-ia64.msi`

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

Install Platform LSF License Scheduler

See *Using Platform LSF License 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.

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.

Special installation steps for the Platform Management Console on Linux IA64

To install the Platform Management Console on Linux IA64 hosts, you must download and install the Linux IA64 version of BEA Jrockit 5.0 JRE.

1 Download the Linux IA64 version of BEA Jrockit 5.0 JRE.

a Open the BEA download page.

`http://commerce.bea.com/products/weblogicjrockit/5.0/jr_50.jsp`

b Save the download file to your local disk.

For JRockit 5.0 R27.1 JRE Linux (Intel Itanium - 64-bit), save the file named `jrockit-R27.1.0-jre1.5.0_08-linux-ipf.bin`.

c Make sure that the `.bin` file is executable.

`chmod +x jrockit-R27.1.0-jre1.5.0_08-linux-ipf.bin`

2 Install the JRE on the Linux IA64 host.

a Change to a shared directory where you want to install BEA Jrockit.

b Run the installer in console mode.

`jrockit-R27.1.0-jre1.5.0_08-linux-ipf.bin -mode=console`

The installation creates a new directory:

`jrockit-R27.1.0-jre1.5.0_08`

3 Follow the steps in *Installing Platform LSF on UNIX and Linux* to run `lsfinstall` to install Platform LSF and the Platform Management Console.

4 Make a symbolic link to the JRE.

For example, if you installed the JRE under `/opt/jre`:

`cd $EG0_TOP/jre`

`ln -s /opt/jre/jrockit-R27.1.0-jre1.5.0_08-linux-ipf linux-ia64`

5 Check the symbolic link to the JRE.

If the symbolic link is correct, you should see the contents of the `linux-ia64` directory:

`cd $EG0_TOP/jre/linux-ia64`

`ls`

`bin/ lib/ LICENSE license.bea README.TXT`

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 Version 7 area of the Platform FTP site (<ftp.platform.com/>).

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, and browse 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. After downloading and extracting the LSF documentation distribution file, browse the file `docs/lsf/7.0/index.html` to access the Platform LSF Knowledge Center.

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

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 `EGO_TOP/docs/ego/1.2.2/index.html`.

If you have installed the Platform Management Console, access the Platform 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.htm 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

supportnotice-request@platform.com with the subject line containing the word "subscribe".

To get security related issue notification from Platform Support, contact

securenotice-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.

Copyright

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

<http://www.platform.com/legal-notice/third-party-license-agreements>