Platform LSF
Version 9 Release 1.2

Release Notes

GI13-3413-03

Platform LSF
Version 9 Release 1.2

Release Notes

GI13-3413-03

Note
Before using this information and the product it supports, read the information in "Notices" on page 31.

First edition

This edition applies to version 9, release 1 of IBM Platform LSF (product number 5725G82) and to all subsequent releases and modifications until otherwise indicated in new editions.

Significant changes or additions to the text and illustrations are indicated by a vertical line (|) to the left of the change.

If you find an error in any Platform Computing documentation, or you have a suggestion for improving it, please let us know. Send your suggestions, comments and questions to the following email address:

pccdoc@ca.ibm.com

Be sure include the publication title and order number, and, if applicable, the specific location of the information about which you have comments (for example, a page number or a browser URL). When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Chapter 1. Release Notes for IBM Platform LSF

Version: 9.1.2

Release date: December 2013

Last modified: November 25, 2013

Support: www.ibm.com/support

Learn more about IBM Platform LSF

Information about IBM Platform LSF (Platform LSF or LSF) is available from the following sources:

- IBM Platform Computing web site: www.ibm.com/systems/technicalcomputing/platformcomputing
- The LSF area of the IBM Support Portal: www.ibm.com/platformcomputing/support.html
- IBM Technical Computing Community on IBM Service Management Connect: www.dw.rtp.raleigh.ibm.com/developerworks/servicemanagement/tc/
- Platform LSF documentation

Platform LSF documentation

The Platform LSF documentation is contained in the LSF documentation packages:

- `lsf9.1.2_documentation.tar.Z`
- `lsf9.1.2_documentation.zip`

You can extract and install these packages to any server on your system. Open the LSF Documentation Center by navigating to the location where you extracted the files and open `index.html` in any Web browser.The Documentation Center provides an overview of the organization of the LSF documentation. It also provides easy access to each document and quick links to frequently used commands and tasks. In addition to links to all documents, the Documentation Center provides full search capabilities within the documentation. You can perform keyword searches within a document or across the full documentation set.

If you have installed IBM Platform Application Center (PAC), you can access and search the LSF documentation through the **Help** link in the user interface.

Platform LSF documentation is also available in PDF format on the IBM Web site:

- By searching the IBM Publications Center: www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss
- By searching the IBM Support Portal: www.ibm.com/support
- On the IBM Cluster Products Information Center: http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp

The documentation set for Platform LSF 9.1.2 includes the following PDF documents:

- *Administering IBM Platform LSF* - SC27530202

- *IBM Platform LSF Foundations* - SC27530402
- *IBM Platform LSF Command Reference* - SC27530502
- *IBM Platform LSF Configuration Reference* - SC27530602
- *Running Jobs with IBM Platform LSF* - SC27530702
- *IBM Platform LSF Quick Reference* - GC27530902
- *Using IBM Platform LSF Advanced Edition* - SC27532102
- *Using IBM Platform LSF on Windows* - SC27531102
- *Using IBM Platform MultiCluster* - SC27531002
- *Installing IBM Platform LSF on UNIX and Linux* - SC27531402
- *Upgrading IBM Platform LSF on UNIX and Linux* - SC27531502
- *Migrating IBM Platform LSF Version 7 to IBM Platform LSF Version 9.1.2 on UNIX and Linux* - SC27531802
- *Installing IBM Platform LSF on Windows* - SC27531602
- *Migrating IBM Platform LSF Version 7 to IBM Platform LSF Version 9.1.2 on Windows* - SC27531702
- *IBM Platform LSF Security* - SC27530302
- *Using IBM Platform LSF with IBM Rational ClearCase* - SC27537700
- *Using the IBM Platform LSF blaunch Framework* - SC27531302
- *IBM Platform LSF Programmer's Guide* - SC27531202

Related documentation can be found in the following 9.1.2 documents:

- *Using IBM Platform License Scheduler* - SC27530802
- *Release Notes for IBM Platform License Scheduler* - GI13341401
- *Using IBM Platform Dynamic Cluster* - SC27532002
- *Release Notes for IBM Platform Dynamic Cluster* - GI13341702
- *IBM Platform MPI User's Guide* - SC27475801
- *Release Notes for IBM Platform MPI: Linux* - GI13189601
- *Release Notes for IBM Platform MPI: Windows* - GI13189701

IBM Technical Computing community

Connect. Learn. Share. Collaborate and network with the IBM Platform Computing experts at the IBM Technical Computing community. Access the Technical Computing community on IBM Service Management Connect at http://www.ibm.com/developerworks/servicemanagement/tc/. Join today!

Service Management Connect is a group of technical communities for Integrated Service Management (ISM) professionals. Use Service Management Connect the following ways:

- Connect to become involved with an ongoing, open engagement among other users, system professionals, and IBM developers of Platform Computing products.
- Learn about IBM Technical Computing products on blogs and wikis, and benefit from the expertise and experience of others.
- Share your experience in wikis and forums to collaborate with the broader Technical Computing user community.

We'd like to hear from you

Contact IBM or your LSF vendor for technical support.

Or go to the IBM Support Portal: www.ibm.com/support

If you find an error in any Platform Computing documentation, or you have a suggestion for improving it, please let us know. Send your suggestions, comments and questions to the following email address:

pccdoc@ca.ibm.com

Be sure include the publication title and order number, and, if applicable, the specific location of the information about which you have comments (for example, a page number or a browser URL). When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Requirements and compatibility

The following sections detail requirements and compatibility for version 9.1.2 of Platform LSF.

System requirements

- IBM AIX 6.*x* and 7.*x* on IBM Power 6/7
- Linux x64 Kernel 2.6 and 3.*x* on x86_64
- Linux Kernel 2.6 and 3.*x* on IBM Power 6/7
- HP UX B.11.31 (64-bit) on HP 9000 Servers (PA-RISC)
- HP UX B.11.31 (IA64) on HP Integrity Servers (Itanium2)
- Solaris 10 and 11 on Sparc
- Solaris 10 and 11 on x86-64
- zLinux Kernal 2.6, glibc 2.4 SLES 10 on S390x-64
- Cray XE6, XT6, XC-30, Linux Kernel 2.6, glibc 2.3 on x86_64
- ARMv7 Kernel 3.6 glibc 2.15 (LSF slave host only)
- Apple Mac OS 10.*x* (LSF slave host only)
- Windows 2003 SP1 and 2, XP SP1 and 2, 2008 x86, 7 X86, and 8 x86 on x86/x86_64
- Windows XP SP1 x64, 2003 SP1/2, 2003 CCE SP1/SP2, 2008 X64, 7 X64, 2008 R2 X64, HPC server 2008, 2012 X64 on x86_64

For detailed LSF system support information, refer to the Compatibility Table at:

http://www.ibm.com/systems/technicalcomputing/platformcomputing/products/lsf/

Master host selection

To achieve the highest degree of performance and scalability, use a powerful master host.

There is no minimum CPU requirement. For the platforms on which LSF is supported, any host with sufficient physical memory can run LSF as master host.

Swap space is normally configured as twice the physical memory. LSF daemons use about 40 MB of memory when no jobs are running. Active jobs consume most of the memory LSF requires.

Cluster size	Active jobs	Minimum recommended memory	Recommended server CPU (Intel, AMD, or equivalent)
Small (<100 hosts)	1,000	1 GB	any server CPU
	10,000	2 GB	recent server CPU
Medium (100-1000 hosts)	10,000	4 GB	multi-core CPU (2 cores)
	50,000	8 GB	multi-core CPU (4 cores)
Large (>1000 hosts)	50,000	16 GB	multi-core CPU (4 cores)
	500,000	32 GB	multi-core CPU (8 cores)

Server host compatibility

Important: To take full advantage of all new features introduced in the latest release of Platform LSF, you *must* upgrade all hosts in your cluster.

Platform LSF 7.x, 8.0.x, 8.3, and 9.1.x, servers are compatible with Platform LSF 9.1.2 master hosts. All LSF 7.x, 8.0.x, 8.3, and 9.1.x features are supported by Platform LSF 9.1.2 master hosts.

LSF Family product compatibility

IBM Platform RTM

Customers can use IBM Platform RTM (RTM) 8.3 or 9.1.x to collect data from Platform LSF 9.1.2 clusters. When adding the cluster, select 'Poller for LSF 8' or 'Poller for LSF 9.1'.

IBM Platform License Scheduler

IBM Platform License Scheduler (License Scheduler) 8.3 and 9.1.x are compatible with Platform LSF 9.1.2.

IBM Platform Analytics

IBM Platform Analytics (Analytics) 8.3 and 9.1.x are compatible with Platform LSF 9.1.2 after the following manual configuration:

To have Analytics 8.3 or 9.1.x collect data from Platform LSF 9.1.2 clusters:

1. Set the following parameters in `lsb.params`:
   - **`ENABLE_EVENT_STREAM=Y`**
   - **`ALLOW_EVENT_TYPE="JOB_NEW JOB_FINISH2 JOB_STARTLIMIT JOB_STATUS2 JOB_PENDING_REASONS"`**
   - **`RUNTIME_LOG_INTERVAL=10`**
2. Copy `elim.coreutil` to LSF:

   ```
   cp ANALYTICS_TOP/elim/os_type/elim.coreutil $LSF_SERVERDIR
   ```
3. In `lsf.shared`, create the following:

```
Begin Resource
RESOURCENAME   TYPE        INTERVAL INCREASING DESCRIPTION
CORE_UTIL      String           300        () (Core Utilization)
End Resource
```

4. In `lsf.cluster.`*`cluster_name`*, create the following:

```
Begin ResourceMap
RESOURCENAME  LOCATION
CORE_UTIL     [default]
End ResourceMap
```

5. Restart all LSF daemons.
6. Configure user group and host group.
7. Run **lsid** and check the output.
8. Install Platform Analytics 8.3 with **COLLECTED_DATA_TYPE=LSF**.
9. Check `perf.conf` to see LSF_VERSION.
10. Restart the Platform loader controller (**plc**).
11. Check the log files and table data to make sure there are no errors.
12. Change all the LSF related data loader intervals to 120 seconds, and run for one day. Check the **plc** and data loader log files to make sure there are no errors.

IBM Platform Application Center

IBM Platform Application Center (PAC) 8.3 and higher versions are compatible with Platform LSF 9.1.*x* after the following manual configuration.

If you are using PAC 8.3 with LSF 9.1.*x*, *$PAC_TOP*`/perf/lsf/8.3` must be renamed to *$PAC_TOP*`/perf/lsf/9.1`

For example:

```
mv /opt/pac/perf/lsf/8.3 /opt/pac/perf/lsf/9.1
```

API compatibility

To take full advantage of new Platform LSF 9.1.2 features, recompile your existing Platform LSF applications with Platform LSF 9.1.2.

Applications need to be rebuilt if they use APIs that have changed in Platform LSF 9.1.2.

New and changed Platform LSF APIs

The following APIs or data structures have changed or are new for LSF Version 9.1.2:

- struct queueInfoEnt
- struct submit_ext
- struct hRusage
- struct jobInfoEnt
- struct jobNewLog
- struct jobModLog
- struct jobFinishLog
- struct jobFinish2Log
- struct jobStatusLog

- struct sbdJobStatusLog
- struct sbdUnreportedStatusLog

For detailed information about APIs changed or created for LSF 9.1.2, refer to the *IBM Platform LSF 9.1.2 API Reference*.

Third party APIs

The following third party APIs have been tested and supported for this release:

- DRMAA LSF API v 1.1.1
- PERL LSF API v1.0
- Python LSF API v1.0 with LSF 9

Packages will be available at www.github.com.

For more information on using third party APIs with LSF 9.1.2 look on the Technical Computing community on IBM Service Management Connect at http://www.ibm.com/developerworks/servicemanagement/tc/.

Installation and migration notes

Upgrade Platform LSF on UNIX and Linux

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

- Upgrade a pre-LSF Version 7 UNIX or Linux cluster to Platform LSF 9.1.*x*
- Upgrade an LSF Version 7 Update 2 or higher to Platform LSF 9.1.*x*

Important: *DO NOT* use the UNIX and Linux upgrade steps to migrate an existing LSF Version 7 or LSF 7 Update 1 cluster to LSF 9.1.*x*. Follow the manual steps in the document *Migrating to Platform LSF Version 9.1.x on UNIX and Linux* to migrate an existing LSF Version 7 or LSF 7 Update 1 cluster to LSF 9.1.*x* on UNIX and Linux.

Migrate LSF Version 7 and Version 7 Update 1 cluster to LSF 9.1.*x* on UNIX and Linux

Follow the steps in *Migrating to Platform LSF Version 9.1.2 on UNIX and Linux* (`lsf_migrate_unix.pdf`) to migrate an *existing* LSF 7 or LSF 7 Update 1 cluster:

- Migrate an existing LSF Version 7 cluster to LSF 9.1.2 on UNIX and Linux
- Migrate an existing LSF Version 7 Update 1 cluster to LSF 9.1.*x* on UNIX and Linux

Note: To migrate an LSF 7 Update 2 or higher cluster to LSF 9.1.*x* follow the steps in *Upgrading IBM Platform LSF on UNIX and Linux*.

Migrate an LSF Version 7 or higher cluster to LSF 9.1.*x* on Windows

To migrate an *existing* LSF 7 Windows cluster to Platform LSF 9.1.2 on Windows, follow the steps in *Migrating IBM Platform LSF Version 7 to IBM Platform LSF Version 9.1.2 on Windows*.

Note: If you want to migrate a pre-version 7 cluster to LSF 9.1.2, you must first migrate the cluster to LSF Version 7.

Platform LSF editions

Platform LSF comes in three editions: Advanced, Standard, and Express.

LSF Advanced Edition

Platform LSF Advanced Edition is architected to support extreme scalability and throughput. LSF Advanced Edition may provide greater than three times more scalability than earlier versions of LSF, enabling you to consolidate your compute resources to achieve maximum flexibility and utilization. LSF Advanced Edition has been tested on configurations up to 18,000 nodes and 160,000 cores running high-throughput workloads of 160,000 concurrent short jobs with 2,000,000 pending jobs. These are not hard scalability or performance limits.

LSF/XL feature

LSF Advanced Edition includes advanced features such as LSF/XL and multithreaded resource requirement matching. IBM Platform LSF Session Scheduler (Session Scheduler) is also included with LSF Advanced Edition.

The LSF/XL feature of LSF Advanced Edition is a new architecture to address long-term scalability and performance demands for extreme workloads. It is based on IBM Platform MultiCluster (Platform MultiCluster) technology, but LSF/XL is not a Platform MultiCluster mode. It is designed for a single data center.

LSF Standard Edition

LSF Standard Edition is designed for grid environments and optimized for complex workloads and user grouping structures. LSF Standard Edition contains full functionality of LSF including functionality for Platform MultiCluster, floating clients and Platform LSF Make. Session Scheduler is available as an add-on component. There are no performance or scalability restrictions.

LSF Standard Edition is subject to the following usage constraints:

- LSF Standard Edition has been tested on clusters up to 6,000 nodes and 60,000 cores running high-throughput workloads of 60,000 concurrent short jobs with 250,000 pending jobs. These are not hard scalability or performance limits. Higher node or core counts can be achieved with a lower volume of jobs such as parallel HPC workloads where cluster sizes of 75,000 to 100,000 cores are supported. Higher core counts are achievable with additional tuning and configuration.
- For high-throughput workloads, the overall system performance depends on the processing power, I/O capacity, and memory of the scheduling node. For very large clusters, seek configuration assistance from IBM.

LSF Express Edition (Linux only)

LSF Express Edition is a solution for Linux customers with very simple scheduling requirements and simple fairshare setup. Smaller clusters typically have a mix of sequential and parallel work as opposed to huge volumes of jobs. For this reason, several performance enhancements and complex scheduling policies designed for large-scale clusters are not applicable to LSF Express Edition clusters. Session Scheduler is available as an add-on component.

Out of box configuration for LSF Express Edition is optimized for smaller clusters with basic scheduling requirements. An LSF Express cluster can have a maximum of 200 server hosts and 200 static client hosts.

LSF Express Edition is subject to the following restrictions:

- LSF Express Edition is supported only on x86_64 Linux.
- LSF Express Edition, LSF Standard Edition, or LSF Advanced Edition cannot coexist in the same cluster.
- Master candidate hosts defined in the **LSF_MASTER_LIST** in `lsf.conf` must exist within the first 200 server hosts in the configuration file. Additional hosts over the limit will not be loaded.

The following LSF Standard Edition features are supported when you upgrade from LSF Express Edition to LSF Standard Edition:

- Job groups
- Live reconfiguration
- Delegation of administrator rights
- Resizable jobs
- Chunk jobs
- Floating clients
- LSF Make
- Resource preemption
- Cross-queue user-based fairshare
- Goal-oriented SLA-driven scheduling
- Guaranteed resource pools
- Slot-based scheduling
- Fairshare scheduling (queue and user fairshare are enabled with a fairshare tree depth of 2 levels in Express and more than two levels in Standard)
- Parallel job scheduling (PAM/PJL is supported. **blaunch** is supported with IBM Platform MPI by default)
- Parallel **mbatchd** restart (**badmin mbdrestart -p**)
- Rapid detection of host failure
- Fast dispatching
- Alternative resource requirement
- **bjobs** shows all levels of resource requirement
- Interaction of select[] and rusage[]
- Process tracking/short jobs
- Platform MultiCluster features
- Multithreaded batch query
- LSF integration with IBM Parallel Environment Runtime Edition
- Memory / CPU enforcement for Linux Cgroup
- Job information security (access control level)
- Energy aware scheduling (CPU frequency control, automatic CPU frequency selection, power state management)
- Global same for compound resource requirements
- Affinity resource preemption
- Host based pre- and post-execution processing

Platform product support with LSF Express Edition

The following IBM Platform products are supported in LSF Express Edition:

- IBM Platform RTM
- IBM Platform Application Center
- IBM Platform License Scheduler

The following IBM Platform products are *not* supported in LSF Express Edition:

- IBM Platform Analytics
- IBM Platform Process Manager

Default configuration for LSF Express Edition

The following table lists the configuration enforced in LSF Express Edition:

Parameter	Setting	Description
RESIZABLE_JOBS in `lsb.applications`	N	If enabled, all jobs belonging to the application will be auto resizable.
EXIT_RATE in `lsb.hosts`	Not defined	Specifies a threshold for exited jobs.
BJOBS_RES_REQ_DISPLAY in `lsb.params`	None	Controls how many levels of resource requirements bjobs –l will display.
CONDENSE_PENDING_REASONS in `lsb.params`	N	Condenses all host-based pending reasons into one generic pending reason.
DEFAULT_JOBGROUP in `lsb.params`	Disabled	The name of the default job group.
EADMIN_TRIGGER_DURATION in `lsb.params`	1 minute	Defines how often LSF_SERVERDIR/eadmin is invoked once a job exception is detected. Used in conjunction with job exception handling parameters JOB_IDLE, JOB_OVERRUN, and JOB_UNDERRUN in lsb.queues.
ENABLE_DEFAULT_EGO_SLA in `lsb.params`	Not defined	The name of the default service class or EGO consumer name for EGO-enabled SLA scheduling.
EVALUATE_JOB_DEPENDENCY in `lsb.params`	Unlimited	Sets the maximum number of job dependencies mbatchd evaluates in one scheduling cycle.
GLOBAL_EXIT_RATE in `lsb.params`	2147483647	Specifies a cluster-wide threshold for exited jobs
JOB_POSITION_CONTROL_BY_ADMIN in `lsb.params`	Disabled	Allows LSF administrators to control whether users can use **btop** and **bbot** to move jobs to the top and bottom of queues.
LSB_SYNC_HOST_STAT_FROM_LIM in `lsb.params`	N	Improves the speed with which **mbatchd** obtains host status, and therefore the speed with which LSF reschedules rerunnable jobs. This parameter is most useful for a large clusters, so it is disabled for LSF Express Edition.

Parameter	Setting	Description
MAX_CONCURRENT_QUERY in `lsb.params`	100	Controls the maximum number of concurrent query commands.
MAX_INFO_DIRS in `lsb.params`	Disabled	The number of subdirectories under the `LSB_SHAREDIR/`*`cluster_name`*`/logdir/info` directory.
MAX_JOBID in `lsb.params`	999999	The job ID limit. The job ID limit is the highest job ID that LSF will ever assign, and also the maximum number of jobs in the system.
MAX_JOB_NUM in `lsb.params`	1000	The maximum number of finished jobs whose events are to be stored in `lsb.events`.
MIN_SWITCH_PERIOD in `lsb.params`	Disabled	The minimum period in seconds between event log switches.
MBD_QUERY_CPUS in `lsb.params`	Disabled	Specifies the master host CPUs on which **mbatchd** child query processes can run (hard CPU affinity).
NO_PREEMPT_INTERVAL in `lsb.params`	0	Prevents preemption of jobs for the specified number of minutes of uninterrupted run time, where minutes is wall-clock time, not normalized time.
NO_PREEMPT_RUN_TIME in `lsb.params`	-1 (not defined)	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.
PREEMPTABLE_RESOURCES in `lsb.params`	Not defined	Enables preemption for resources (in addition to slots) when preemptive scheduling is enabled (has no effect if queue preemption is not enabled) and specifies the resources that will be preemptable.
PREEMPT_FOR in `lsb.params`	0	If preemptive scheduling is enabled, this parameter is used to disregard suspended jobs when determining if a job slot limit is exceeded, to preempt jobs with the shortest running time, and to optimize preemption of parallel jobs.
SCHED_METRIC_ENABLE in `lsb.params`	N	Enables scheduler performance metric collection.
SCHED_METRIC_SAMPLE_PERIOD in `lsb.params`	Disabled	Performance metric sampling period.
SCHEDULER_THREADS in `lsb.params`	0	Sets the number of threads the scheduler uses to evaluate resource requirements.

Parameter	Setting	Description
DISPATCH_BY_QUEUE in `lsb.queues`	N	Increases queue responsiveness. The scheduling decision for the specified queue will be published without waiting for the whole scheduling session to finish. The scheduling decision for the jobs in the specified queue is final and these jobs cannot be preempted within the same scheduling cycle.
LSB_JOBID_DISP_LENGTH in `lsf.conf`	Not defined	By default, LSF commands bjobs and bhist display job IDs with a maximum length of 7 characters. Job IDs greater than 9999999 are truncated on the left. When LSB_JOBID_DISP_LENGTH=10, the width of the JOBID column in bjobs and bhist increases to 10 characters.
LSB_FORK_JOB_REQUEST in `lsf.conf`	N	Improves **mbatchd** response time after **mbatchd** is restarted (including parallel restart) and has finished replaying events.
LSB_MAX_JOB_DISPATCH_PER_SESSION in `lsf.conf`	300	Defines the maximum number of jobs that **mbatchd** can dispatch during one job scheduling session.
LSF_PROCESS_TRACKING in `lsf.conf`	N	Tracks processes based on job control functions such as termination, suspension, resume and other signaling, on Linux systems which support cgroups' freezer subsystem.
LSB_QUERY_ENH in `lsf.conf`	N	Extends multithreaded query support to batch query requests (in addition to **bjobs** query requests). In addition, the **mbatchd** system query monitoring mechanism starts automatically instead of being triggered by a query request. This ensures a consistent query response time within the system. Enables a new default setting for *min_refresh_time* in MBD_REFRESH_TIME (`lsb.params`).
LSB_QUERY_PORT in `lsf.conf`	Disabled	Increases **mbatchd** performance when using the **bjobs** command on busy clusters with many jobs and frequent query request.
LSF_LINUX_CGROUP_ACCT in `lsf.conf`	N	Tracks processes based on CPU and memory accounting for Linux systems that support cgroup's memory and cpuacct subsystems.

IBM Platform entitlement files

Entitlement files are used for determining which edition of the product is enabled. The following entitlement files are packaged for LSF:

- LSF Standard Edition: `platform_lsf_std_entitlement.dat`
- LSF Express Edition: `platform_lsf_exp_entitlement.dat`
- LSF Advanced Edition: `platform_lsf_adv_entitlement.dat`

The entitlement file for the edition you use is installed as `LSF_TOP/conf/lsf.entitlement`.

If you have installed LSF Express Edition, you can upgrade later to LSF Standard Edition or LSF Advanced Edition to take advantage of the additional functionality. Simply reinstall the cluster with the LSF Standard entitlement file (`platform_lsf_std_entitlement.dat`) or the LSF Advanced entitlement file (`platform_lsf_adv_entitlement.dat`).

You can also manually upgrade from LSF Express Edition to Standard Edition or Advanced Edition. Get the LSF Standard or Advanced Edition entitlement file, copy it to `LSF_TOP/conf/lsf.entitlement` and restart you cluster. The new entitlement enables the additional functionality of LSF Standard Edition, but you may need to manually change some of the default LSF Express configuration parameters to use the LSF Standard or Advanced features.

To take advantage of LSF SLA features in LSF Standard Edition, copy `LSF_TOP/LSF_VERSION/install/conf_tmpl/lsf_standard/lsb.serviceclasses` into `LSF_TOP/conf/lsbatch/LSF_CLUSTERNAME/configdir/`.

Once LSF is installed and running, run the **lsid** command to see which edition of LSF is enabled.

What's new in Platform LSF Version 9.1.2

New and changed behavior

Job information security

LSF has features for controlling job security. This allows you to set the access control level to jobs by regular users and administrators. This is useful for large environments where many groups may share the same cluster and it may be a security threat to allow some users to view job details and summary information. With access control levels configured, you may prevent users (including user group, queue, and cluster administrators) from viewing other user's job information through LSF commands (such as **bjobs**, **bjdepinfo**, **bread**, **bstatus**, **bhist**, and **bacct**).

There are two kinds of job information which will be viewed by users:

- Summary Information:

  Obtained from **bjobs** with options other than **-l**, such as **-aps, -fwd, -p, -ss, -sum, -W, -WF, -WP, -WL,** etc.
- Detail Information:

  Obtained from **bjobs -l**, **bjobs -UF**, **bjobs -N**, **bjdepinfo**, **bread**, and **bstatus**.

There are three parameters available in `lsb.params` that allow you to control access to job information: **SECURE_JOB_INFO_LEVEL**, **ENABLE_JOB_INFO_BY_ADMIN_ROLE**, and **SECURE_INFODIR_USER_ACCESS**.

The parameter **SECURE_JOB_INFO_LEVEL** in `lsb.params` allows you to define an access control level for all users (including user group, queue, and cluster administrators). A value between 0 and 4 is defined, with 0 being no security and 4 being the highest security.

When a user enters one of the commands to see job information (**bjobs**, **bjdepinfo**, **bread**, or **bstatus**), the **SECURE_JOB_INFO_LEVEL** controls whether they see:

- Just their own jobs' information. (level 4)
- Their own jobs and summary information from jobs in the same user group. (level 3)
- Their own jobs, summary and detail information from jobs in the same user group. (level 2)
- Their own jobs, summary and detail information from jobs in the same user group, and summary information from jobs outside their user group. (level 1)
- Summary and detail job information for all jobs. (level 0)

By default, an administrator's access to job details is determined by the setting of **SECURE_JOB_INFO_LEVEL**, the same as a regular user. The parameter **ENABLE_JOB_INFO_BY_ADMIN_ROLE** in `lsb.params` allows you to enable the user group, queue, and cluster administrators the right to access job detail information for jobs in the user group, queue, and clusters they manage, even when the administrator has no right based on the configuration of **SECURE_JOB_INFO_LEVEL**.

Note: This does not apply to the primary administrator or root.

The parameter **SECURE_INFODIR_USER_ACCESS** in `lsb.params` allows you to control whether regular and administrator users (except the primary admin) can see other user's jobs when using the **bhist** or **bacct** command.

If enabled (defined as `Y`), regular users and administrators can view only their own job information when using the **bhist** or **bacct** command. `LSB_SHAREDIR/cluster/logdir` is readable only by the Primary Administrator.

When disabled (defined as `N`), access to read `LSB_SHAREDIR/cluster/logdir` returns to default after an **mbatchd restart** or **reconfig**.

Attention: Requirements

An upgrade to LSF 9.1.2 is required for this feature.

After enabling this feature, you must **setuid** of the LSF primary administrator for **bhist** and **bacct** binary under **LSF_BINDIR**. **bhist** and **bacct** will call **mbatchd** to check whether the parameter is set or not when you have **setuid** for **bhist** and **bacct**.

Note: When job information security is enabled, pre-LSF 9.1 **bjobs** and **bjdepinfo** commands will be rejected no matter who issues them because mbatchd cannot get the command user name. A "No job found" message will be returned.

Note: Some batch commands (**bkill**, **bstop**, **bresume**, **bchkpnt**, **bmig**, **brequeue**, **bswitch**) which use job query API will be affected by ACL (job security) feature.

Global Same for compound resource requirements

A "global same" has been introduced as part of a job's resource requirement expression, to take effect over multiple component subexpressions of a compound resource requirement string. This new functionality can be used in any compound resource requirement, regardless of whether specified at the job, application, or queue level.

The "same" requirement can be used within a resource requirement expression of a parallel job, to ensure that all hosts allocated to the job share some common attributes. For example, one might submit a parallel job as:

```
bsub -n 128 -R "same[type]" ./a.out
```

In this case, LSF will allocate 128 slots to the job, all on hosts of the same type.

The "same" requirement also works for user defined resources. For example, you might configure a string resource called "rack", and set the value of this resource on each host to the ID of the rack that contains the host. Then, a job may be submitted as:

```
bsub -n 128 -R "same[rack]" ./a.out
```

In this case, LSF will allocate 128 slots to this job, all on hosts within the same rack.

Compound resource requirements allow a single job to have different resource requirements on different parts of its allocation. This is useful in clusters, for example, where some hosts are dedicated to IO functions, while other hosts are dedicated to compute. Some jobs may require amounts of each type, such as in the following example:

```
bsub -R "16*{type=io} + 128*{type=compute}" ./a.out
```

This job requests 16 slots on IO hosts, and 128 slots on compute hosts.

Previously, LSF had the limitation that the "same" string could only be specified within the simple subexpressions of a compound resource requirement. The "same" would take effect on the set of hosts allocated for each subexpression, but not for the allocation as a whole. In the above job submission, it would not be possible to use the "same" requirement to ensure that all slots come from a single rack.

In this release, we remove this limitation. Now, the "same" requirement can be applied to multiple subexpressions of a compound resource requirement expression:

```
bsub -R "{16*{type=io} + 128*{type=compute}} same[rack] " ./a.out
```

This job requests 16 slots on IO hosts, and 128 slots on compute slots. All slots allocated to the job must come from hosts the same rack.

Installer configuration templates and initial settings

When installing LSF Standard Edition on UNIX or Linux, you may select a configuration template that specifies initial configuration parameter values. This

allows you to specify an initial setup that is appropriate for the specific type of cluster you are installing, depending on its purpose.

Note: These configuration templates are not available with LSF Advanced Edition.

To select a configuration template, edit `install.config` and uncomment the **CONFIGURATION_TEMPLATE** parameter:

```
CONFIGURATION_TEMPLATE="DEFAULT" | "PARALLEL" | "HIGH_THROUGHPUT"
```

The following are the valid values for this parameter:

DEFAULT
: This template should be used for clusters with mixed workload. This configuration can serve different types of workload with good performance, but is not specifically tuned for a particular type of cluster.

PARALLEL
: This template provides extra support for large parallel jobs. This configuration is designed for long running parallel jobs, and should not be used for clusters that mainly run short jobs due to the longer reporting time for each job.

HIGH_THROUGHPUT
: This template is designed to be used for clusters that mainly run short jobs, where over 80% of jobs finish within one minute. This high turnover rate requires LSF to be more responsive and fast acting. However, this configuration will consume more resources as the daemons become busier.

If you do not select a configuration template, the DEFAULT configuration template is selected by default.

The installer uses the DEFAULT configuration template when installing LSF Standard Edition on Windows.

Note: Do not specify **CONFIGURATION_TEMPLATE** for LSF Express Edition and Advanced Edition. These editions have their own default configuration templates for all installations.

The installer specifies the following initial configuration file parameter values based on the selected configuration template:

- DEFAULT
  - `lsf.conf`:

    ```
    DAEMON_SHUTDOWN_DELAY=180
    LSF_LINUX_CGROUP_ACCT=Y
    LSF_PROCESS_TRACKING=Y
    ```

  - `lsb.params`:

    ```
    JOB_DEP_LAST_SUB=1
    JOB_SCHEDULING_INTERVAL=1
    MAX_JOB_NUM=10000
    NEWJOB_REFRESH=Y
    SBD_SLEEP_TIME=7
    ```

- PARALLEL
  - `lsf.conf`:

```
LSB_SHORT_HOSTLIST=1
LSF_LINUX_CGROUP_ACCT=Y
LSF_PROCESS_TRACKING=Y
LSF_ENABLE_EXTSCHEDULER=Y
LSF_HPC_EXTENSIONS="CUMULATIVE_RUSAGE LSB_HCLOSE_BY_RES SHORT_EVENTFILE"
```

Refer to the Enable LSF HPC Features section for a full description.

– `lsb.params`:

```
JOB_DEP_LAST_SUB=1
JOB_SCHEDULING_INTERVAL=1
NEWJOB_REFRESH=Y
```

- HIGH_THROUGHPUT
  – `lsf.conf`:

```
LSB_MAX_PACK_JOBS=300
LSB_SHORT_HOSTLIST=1
```

  – `lsb.params`:

```
CONDENSE_PENDING_REASONS=Y
JOB_SCHEDULING_INTERVAL=50ms
MAX_INFO_DIRS=500
MAX_JOB_ARRAY_SIZE=10000
MAX_JOB_NUM=100000
MIN_SWITCH_PERIOD=1800
NEWJOB_REFRESH=Y
PEND_REASON_UPDATE_INTERVAL=60
SBD_SLEEP_TIME=3
```

The installer specifies the following initial configuration parameters for all configuration templates:

- `lsf.conf`:

```
EGO_ENABLE_AUTO_DAEMON_SHUTDOWN=Y
LSB_DISABLE_LIMLOCK_EXCL=Y
LSB_MOD_ALL_JOBS=Y
LSF_DISABLE_LSRUN=Y
LSB_SUBK_SHOW_EXEC_HOST=Y
LSF_PIM_LINUX_ENHANCE=Y
LSF_PIM_SLEEPTIME_UPDATE=Y
LSF_STRICT_RESREQ
LSF_UNIT_FOR_LIMITS=MB
```

- `lsb.params`:

```
ABS_RUNLIMIT=Y
DEFAULT_QUEUE=normal interactive
JOB_ACCEPT_INTERVAL=0
MAX_CONCURRENT_JOB_QUERY=100
MBD_SLEEP_TIME=10
PARALLEL_SCHED_BY_SLOT=Y
```

In addition, the installer enables the following features for all configuration templates:

- Fairshare scheduling (LSF Standard Edition and Advanced Edition): All queues except `admin` and `license` have fairshare scheduling enabled as follows in `lsb.queues`:

```
Begin Queue
...
FAIRSHARE=USER_SHARES[[default, 1]]
...
End Queue
```

- Host groups (LSF Standard Edition on UNIX or Linux): Master candidate hosts are assigned to the `master_hosts` host group.

- User groups (LSF Standard Edition on UNIX or Linux): LSF administrators are assigned to the `lsfadmins` user group.
- Affinity scheduling in both `lsb.modules` and `lsb.hosts`.

LSF event streaming

You can now enable LSF event streaming during installation by specifying `ENABLE_STREAM="Y"` in `install.config` before running the LSF installer.

Enable LSF event streaming if you intend to install IBM Platform Analytics or IBM Platform Application Center.

Block scheduling

For applications that are not especially sensitive to network latency, or where you prefer to get throughput, you can allocate slots for a parallel job with a specific block size. The applications specified by the job may be running as threaded processes on groups of n cores, but using MPI applications or other socket connections between blocks. LSF will allocate slots to the job based on block size. LSF tries to pack as many blocks on one host as possible, then goes to next one. Each host is only checked once. It does not matter which host contains the slot blocks. The job can start as soon as any previous job is complete.

This packing policy is supported by the keyword block (`"span[block=value]"`) in the span section of the resource requirement string. `"span[block=value]"` can also be configured in the **`RES_REQ`** parameter in `lsb.queues` and `lsb.applications`.

When a block size is specified for a job, LSF allocates only a multiple of the block size for the job. The minimum and maximum values in **`-n min,max`** are also adjusted to be a value of multiple of the block.

Define GPU or MIC resources

You can enable LSF so applications can use Nvidia Graphic Processing Units (GPUs) or Intel MIC (Phi co-processors) in a Linux environment. LSF supports parallel jobs that request GPUs or MICs, allowing you to specify a certain number of GPUs or MICs on each node at run time, based on availability.

Specifically, LSF 9.1.2 supports the following:

- Nvidia GPUs for serial and parallel jobs. Parallel jobs should be launched by **`blaunch`**.
- Intel MIC (Phi co-processor) for LSF jobs in offload mode, both serial and parallel.
- CUDA 4.0 to CUDA 5.5.
- Linux x64: MIC supports Linux x64. Linux-based GPUs support x64 for RHEL/Fedora/SLES.

LSF also supports the collection of metrics for GPUs and MICs using elims and predefined LSF resources.

The previous GPU package is replaced by the new design for LSF 9.1.2. If you want to use the previous GPU package, do the following:

For an upgrade install:

1. Replace `elim.gpu` in `$LSF_SERVERDIR` with the old GPU elim.
2. Keep the old configuration files and do not use the LSF9.1.2 GPU related resource definition.
3. Restart the cluster

For a new install:

1. Replace `elim.gpu` in `$LSF_SERVERDIR` with the old GPU elim.
2. Define GPU related resources as the old solution.
3. Restart the cluster.

Host based pre- and post-execution processing

LSF previously featured job-based pre- and post-execution processing which was intended for sequential jobs, and where pre- and post-execution processing ran only on the first execution host. For release 9.1.2, LSF features host-based pre- and post-execution processing, which is intended for parallel jobs (you can also use this feature for sequential jobs) and runs on all execution hosts. The purpose of this is to set up the execution hosts before all job-based pre-execution and other pre-processing which depend on host-based preparation, and clean up execution hosts after job-based post execution and other post-processing.

This feature can be used in a number of ways. For example:

- HPC sites can have multiple ways to check for system health before actually launching jobs, such as checking for host or node status, key file systems are mounted, infiniband is working, required directories, files, environment, and correct user permissions are set, etc.)
- Administrators can configure site specific policy to run host-based pre- and post-execution processing to set up ssh access to computer nodes. By default, ssh is disabled. However, with host-based pre- and post-execution processing, ssh access to the nodes allocated for the job can be enabled for the duration of job life cycle. This is required for debugging a parallel job on a non-first execution host and will not impact the overall cluster security policy.
- Administrators can configure host-based pre- and post-execution processing to create and later remove temporary working directories on each host.

The following configuration parameters can be used for both job-based and host-based pre- and post-execution processing:

- **JOB_PREPROC_TIMEOUT**
- **JOB_POSTPROC_TIMEOUT**
- **LSB_PRE_POST_EXEC_USER**
- **LSB_POSTEXEC_SEND_MAIL**
- **JOB_INCLUDE_POSTPROC**
- **LOCAL_MAX_PREEXEC_RETRY**
- **MAX_PREEXEC_RETRY**
- **REMOTE_MAX_PREEXEC_RETRY**
- **LSB_DISABLE_RERUN_POST_EXEC**

Kerberos Support for NFSv4 and AFS

When using LSF on NFSv4 or Andrew File System (AFS), each process in a sequential job or a distributed parallel job needs to periodically renew its

credentials. For this re-authentication to take place in a secure, user friendly environment, a TGT file is distributed to each execution host and the root sbatchd in each execution host renews the TGT.

To support AFS, LSF provides an external renew hook mechanism which is called after TGT is renewed. Users can write their own renew logic through this renew hook. More specifically, users can use the demo script named `erenew.krb5` in `$LSF_SERVERDIR` and rename it to **`erenew`**. Users can also create an executable named **`erenew`** in `$LSF_SERVERDIR`. This **`erenew`** script will be called immediately at job startup time to make sure the user's job has a valid AFS token. LSF will also automatically call this binary after TGT is renewed. For example, AFS users can use this hook to run `aklog` for renewing AFS tokens.

Note: blaunch krb5 does not support pre LSF 9.1.2 remote execution server, and therefore the renew script will not work in pre 9.1.2 RES. Similarly, blaunch krb5 does not support pre LSF 9.1.2 sbatchd. Therefore, child sbatchds cannot be kerberized and the renew script does not work in pre 9.1.2 root sbatchd

Note: No krb on Solaris platforms for LSF 9.1.2.

Energy Aware Scheduling

LSF offers energy-aware scheduling features for large-scale LSF installations, where the energy requirements for operating large systems are becoming a significant factor in the overall cost of these systems. On large systems with either a long lead period to full production or widely fluctuating workloads many nodes can sit idle for significant time periods. The energy-aware scheduling features of LSF enable administrators to control the processor frequency to allow some applications to run at lower frequency with minor performance degradation. This can lead to overall power savings. Conversely, minimizing the frequency on unused cores can also enable maximum turbo boost to active cores, to increase application performance, and reduce run times. Frequency control allows an organization to balance performance with power savings. As well, manual or policy controlled power management is available to cluster administrators to suspend (S3) and resume on the specified hosts or host groups through LSF command line.

In contradiction, if job runtime is more important than energy savings, the CPU frequency can be increased per job.

LSF energy-aware scheduling features include the following:

- Host-based policies to manage the power state of hosts.
- Ability to set the CPU frequency at the job, application, or user level.
- Collection and reporting of power usage for an application (assuming exclusive use of nodes).
- Benchmarking application power usage and generation of relevant power coefficients.
- Prediction of performance, power usage, and runtime of applications at different CPU frequencies.
- Automatic CPU frequency selection for jobs based on predictions.

Energy aware scheduling features are available only for LSF Standard Edition.

Energy aware scheduling features have some extra installation and configuration requirements and dependency on third party tools. For example:

- STREAM and NPB-NAS Parallel Benchmarks
- MySQL DB or xCat MySQL database
- mysql-connector-odbc or unixODBC must be on the master/master candidate hosts
- cpufrequtils package is installed on all compute nodes
- The following Linux kernel modules must be installed on all nodes: msr, ibmaem, ipmi_si, acpi_cpufreq
- All compute nodes have P-States and C-States enabled
- IBM iDataplex is supported, on homogeneous nodes (same hardware, OS, CPU count, memory)
- Hyperthreading must be disabled on all nodes
- No compute node may be in turbo-boost mode.

For information on system requirements for energy aware scheduling features, see the Energy Aware Scheduling chapter in *Administering IBM Platform LSF*.

License Scheduler Basic Edition available with LSF

License Scheduler Basic Edition is now available for use with LSF 9.1.2 at no additional charge.

License Scheduler Basic Edition monitors the availability of licenses managed by FlexNet, and throttles the workload of a single cluster to avoid dispatching more jobs with license requirements than can run with the available licenses. It can also be used to view the license use of individual jobs.

You can replace an elim that tracks licenses managed by FlexNet with License Scheduler Basic Edition.

Note that License Scheduler Basic Edition is intended for use only in cases where licenses are not shared by multiple clusters. It does not dynamically balance license use among multiple clusters, or among multiple projects. In cases where such functionality is required, use License Scheduler Standard Edition.

To install and run License Scheduler Basic Edition, download and install the License Scheduler packages as described in *Using IBM Platform License Scheduler* (specifically, in the *Installing and starting License Scheduler* section), but follow any specific steps for installing and configuring License Scheduler Basic Edition instead of Standard Edition.

Changes to default LSF behavior

With no new features enabled in a newly upgraded LSF 9.1.*x* cluster, the following pre-9.1 functionality has changed:

- When installing LSF Standard Edition on UNIX or Linux, you may select a configuration template by specifying the **CONFIGURATION_TEMPLATE** parameter in the `install.config` file. The LSF installer sets initial configuration parameters based on the configuration template that you select. For more details, refer to "Installer configuration templates and initial settings" on page 14.
- By default, **bjobs -l** reports individual host rusage for a parallel job. Set **LSF_HPC_EXTENSIONS=NO_HOST_RUSAGE** to enable pre-LSF 9.1.*x* behavior.
- **bswitch** does not modify the effective resource requirement of a running job based on the resource requirements of the destination queue res req definition. You can define **BSWITCH_MODIFY_RUSAGE** to enable pre-LSF 9.1.*x* behavior.

- LSF now uses non-privileged ports by default for daemon communication. You can set **LSF_NON_PRIVILEGED_PORTS=N** in `lsf.conf` to enable privileged port communication. Also, **LSF_MC_NON_PRIVILEGED_PORTS** and **LSF_NON_PRIVILEGED_PORTS** are now fully decoupled, which is different from previous versions.

If you are upgrading your master host and leaving some server hosts still running older versions, do the following:

- If LSF_NON_PRIVILEGED_PORTS is already set to Y or N, continue with upgrade.
- If LSF_NON_PRIVILEGED_PORTS is not set, but **LSB_MAX_JOB_DISPATCH_PER_SESSION** is set to a value greater than 300, do the following:
  1. Shut down the cluster
  2. Set LSF_NON_PRIVILEGED_PORTS=Y
  3. Upgrade the master host
  4. Restart the cluster
- If neither LSF_NON_PRIVILEGED_PORTS nor LSB_MAX_JOB_DISPATCH_PER_SESSION is set, do the following:
  1. Set LSF_NON_PRIVILEGED_PORTS=N.
  2. Upgrade the master host.
  3. Start LSF on the master host.

See *Upgrading IBM Platform LSF on UNIX and Linux* for detailed upgrade steps.

New commands

New commands added for LSF 9.1.2.

- **bentags**: Used with energy policy, or the energy aware scheduling feature. The bentags command queries or removes information about the energy policy tag from mbatchd which is saved in the database. This command displays all the energy tag names that have been generated by the user and can remove energy policy tags.

New configuration files

New configuration file added for LSF 9.1.2.

- **lsb.threshold**: To enable the automatic select CPU frequency feature of energy aware scheduling, you must define the lsb.threshold configuration file, using the energy tags (accessed using **bentags**).

  The lsb.threshold file is available at the location specified by the parameter PERFORMANCE_THRESHOLD_FILE in lsb.params. The default location is $LSF_ENVDIR/lsbatch/cluster_name/configdir/lsb.threshold.

New and changed commands, options, and output

The following command options and output are new or changed for LSF 9.1.2:

badmin

- New subcommand option `hpower` is used to manually switch hosts between a power saving state **suspend** or a working state **resume**.
- Subcommand options `hist` and `hhist` can be used to retrieve a host's history of power state changes. Both badmin and policy (job)-triggered power related events are logged as type **HOST_POWER_STATUS**.

bapps

- **HOST_PRE_EXEC**: The host based pre-execution command for the application profile. The **HOST_PRE_EXEC** command runs on all execution hosts before the job associated with the application profile is dispatched to the execution hosts. If job based pre-execution **PRE_EXEC** was defined at the queue-level/application-level/job-level, the **HOST_PRE_EXEC** command runs before **PRE_EXEC** of any level. The host-based pre-execution command cannot be executed on Windows platforms.
- **HOST_POST_EXEC**: The post-execution command for the application profile. The **HOST_POST_EXEC** command runs on all execution hosts after the job finishes. If job based post-execution **POST_EXEC** was defined at the queue-level/application-level/job-level, the **HOST_POST_EXEC** command runs after **POST_EXEC** of any level. The host-based post-execution command cannot be executed on Windows platforms.

bhist

- The `-t` option displays job events chronologically, including new events for energy aware scheduling, JOB_PROV_HOST and HOST_POWER_STATUS.

bhosts

- The `-l` option displays host power states when PowerPolicy is enabled (in lsb.resources). Final power states are **on** or **suspend**. Intermediate power states are **restarting**, **resuming**, and **suspending**. If the host batch status becomes unknown (power operation due to failure), the power state is shown as a dash ("-").

bjobs

- PROV has been added as a possible value for JOB STATUS in the long format output (-l). This status means the job has been dispatched to a power-saved host that is waking up. Before the job can be sent to the sbatchd, it is in a PROV state.
- If the job was submitted with an energy policy, to automatically select a CPU frequency, `-l` will show the Combined CPU frequency (the CPU frequency selected for the job based on the energy policy tag, energy policy and threshold file). If the job was submitted with a user defined CPU frequency (using **bsub –freq**), `-l` will show the Specified CPU frequency for the job.

bmod

- The -freq option specifies a CPU frequency for a job. The submission value will overwrite the application profile value and the application profile value will overwrite the queue value. The value is float and should be specified with SI units (GHz, MHz, KHz), for example `bmod -freq 2.5GHz`. If no units are specified, GHz is the default.

bqueues

- **HOST_PRE_EXEC**: The host based pre-execution command for the queue. The **HOST_PRE_EXEC** command runs on all execution hosts before the job associated with the queue is dispatched to the execution hosts. If job based pre-execution **PRE_EXEC** was defined at the queue-level/application-level/job-level, the **HOST_PRE_EXEC** command runs before **PRE_EXEC** of any level. The host-based pre-execution command cannot be executed on Windows platforms.
- **HOST_POST_EXEC**: The post-execution command for the queue. The **HOST_POST_EXEC** command runs on all execution hosts after the job finishes. If job based post-execution **POST_EXEC** was defined at the queue-level/application-

level/job-level, the **HOST_POST_EXEC** command runs after **POST_EXEC** of any level. The host-based post-execution command cannot be executed on Windows platforms.

bresources

- The `-p` option displays the currently defined energy aware scheduling policies and exits. Shows the PowerPolicy settings as they are in lsb.resources. An additional line is included with the PowerPolicy settings to indicate whether it is currently Applied (Y) or not (N).

bsub

- The -freq option specifies a CPU frequency for a job. The submission value will overwrite the application profile value and the application profile value will overwrite the queue value. The value is float and should be specified with SI units (GHz, MHz, KHz), for example `bsub -freq 2.5GHz`. If no units are specified, GHz is the default.

New and changed configuration parameters and environment variables

install.config

- **CONFIGURATION_TEMPLATE**: Selects the configuration template for this installation, which determines the initial LSF configuration parameters specified when the installation is complete.

  ```
  CONFIGURATION_TEMPLATE="DEFAULT" | "PARALLEL" | "HIGH_THROUGHPUT"
  ```

- **ENABLE_STREAM**: Enables LSF event streaming for Analytics or PAC.

  ```
  ENABLE_STREAM="Y" | "N"
  ```

lsb.applications

- **HOST_PRE_EXEC**: Enables host-based pre-execution processing at the application level. The **HOST_PRE_EXEC** command runs on all execution hosts before the job starts. If job based pre-execution **PRE_EXEC** was defined at the queue-level/application-level/job-level, the **HOST_PRE_EXEC** command runs before **PRE_EXEC** of any level. **HOST_PRE_EXEC** is not supported on Windows platforms.

  ```
  HOST_PRE_EXEC=command
  ```

- **HOST_POST_EXEC**: Enables host-based post-execution processing at the queue level. The **HOST_POST_EXEC** command runs on all execution hosts after the job finishes. If job based pre-execution **POST_EXEC** was defined at the queue-level/application-level/job-level, the **HOST_POST_EXEC** command runs after **POST_EXEC** of any level. **HOST_POST_EXEC** is not supported on Windows platforms.

  ```
  HOST_POST_EXEC=command
  ```

- **CPU_FREQUENCY**: Specifies the CPU frequency for an application profile. All jobs submit to the application profile require the specified CPU frequency. Value is a positive float number with units (GHz, MHz, or KHz). If no units are set, the default is GHz. This value can also be set using the command **bsub –freq**. The submission value will overwrite the application profile value, and the application profile value will overwrite the queue value.

  ```
  CPU_FREQUENCY=[float_number][unit]
  ```

lsb.params

- **SECURE_JOB_INFO_LEVEL**: Defines which jobs all users can see information for. A value between 0 and 4 is defined, with 0 being no security and 4 being the highest security.

SECURE_JOB_INFO_LEVEL=0|1|2|3|4

- **ENABLE_JOB_INFO_BY_ADMIN_ROLE**: Enables user group, queue, and cluster administrators the right to access job detail information for jobs in the user group, queue, and clusters they manage, even when the administrator has no right based on the configuration of **SECURE_JOB_INFO_LEVEL**.You may define one or more of the values, usergroup, queue, or cluster.

  ENABLE_JOB_INFO_BY_ADMIN_ROLE=[usergroup] [queue] [cluster]

- **SECURE_INFODIR_USER_ACCESS**: Controls whether regular users can see other user's jobs when using the bhist or bacct command. If enabled (defined as Y), the primary administrator will still be able to view all job information in lsb.event and lsb.acct.

  SECURE_INFODIR_USER_ACCESS=Y | N

- **MAX_JOB_PREEMPT_RESET**: Does not reset the preempted count for **MAX_JOB_PREEMPT** when the started job is requeued, migrated or rerun in SSUSP state.

  MAX_JOB_PREEMPT_RESET=Y|N

- **POWER_ON_WAIT**: Configures a wait time (in seconds) after a host is resumed and enters ok status, before dispatching a job. This is to allow other services on the host to restart and enter a ready state. The default value is 0 and is applied globally.

  POWER_ON_WAIT=*time_seconds*

- **POWER_RESUME_CMD**: Defines the resume operation script that will be called when handling a power resume request.

  POWER_RESUME_CMD=*command*

- **POWER_RESET_CMD**: Defines the reset operation script that will be called when handling a power reset request.

  POWER_RESET_CMD=*command*

- **POWER_STATUS_LOG_MAX**: Configures a trigger value for events switching. The default value is 10000. This value takes effect only if PowerPolicy (in lsb.resources) is enabled. If a finished job number is not larger than the value of **MAX_JOB_NUM**, the event switch can also be triggered by POWER_STATUS_LOG_MAX, which works with MIN_SWITCH_PERIOD. Not available with LSF Express edition.

  POWER_STATUS_LOG_MAX=*number*

- **POWER_SUSPEND_CMD**: Defines the suspend operation script that will be called when handling a power suspend request.

  POWER_SUSPEND_CMD=*command*

- **POWER_SUSPEND_TIMEOUT**: Defines the timeout value (in seconds) for power suspend, resume, and reset actions. When a power operation is not successful (for example, sbatchd does not reconnect when resuming a host) within the specified number of seconds, the action will be considered failed.

  POWER_SUSPEND_TIMEOUT=*integer*

- **PERFORMANCE_THRESHOLD_FILE**: Specifies the location of the performance threshold file for the cluster. This file contains the cluster-level threshold values for the minimize energy and minimize time policies used for automatic CPU frequency selection.

  PERFORMANCE_THRESHOLD_FILE=*full_file_path*

lsb.queues

- **HOST_PRE_EXEC**: Enables host-based pre-execution processing at the queue level. The **HOST_PRE_EXEC** command runs on all execution hosts before the job starts. If job based pre-execution **PRE_EXEC** was defined at the queue-level/application-

level/job-level, the**HOST_PRE_EXEC** command runs before **PRE_EXEC** of any level. **HOST_PRE_EXEC** is not supported on Windoows platforms.

```
HOST_PRE_EXEC=command
```

- **HOST_POST_EXEC**: Enables host-based post-execution processing at the queue level. The **HOST_POST_EXEC** command runs on all execution hosts after the job finishes. If job based pre-execution **POST_EXEC** was defined at the queue-level/application-level/job-level, the**HOST_POST_EXEC** command runs after **POST_EXEC** of any level. **HOST_POST_EXEC** is not supported on Windoows platforms.

  ```
  HOST_POST_EXEC=command
  ```

- **CPU_FREQUENCY**: Specifies the CPU frequency for a queue. All jobs submit to the queue require the specified CPU frequency. Value is a positive float number with units (GHz, MHz, or KHz). If no units are set, the default is GHz. This value can also be set using the command **bsub –freq**. The submission value will overwrite the application profile value, and the application profile value will overwrite the queue value.

  ```
  CPU_FREQUENCY=[float_number][unit]
  ```

lsb.resources

- **PowerPolicy** section: Enables and defines a power management policy.

lsf.conf

- **LSB_AFS_BIN_DIR**: If **LSB_AFS_JOB_SUPPORT=Y**, then LSF will need `aklog` in AFS to create a new PAG and apply for an AFS token. You can then use **LSB_AFS_BIN_DIR** to tell LSF the file path and directory where `aklog` resides. If **LSB_AFS_BIN_DIR** is not defined, LSF will search in the following order: `/bin`, `/usr/bin`, `/usr/local/bin`, `/usr/afs/bin`. The search stops as soon as an executable `aklog` is found.

  ```
  LSB_AFS_BIN_DIR=path to aklog directory
  ```

- **LSB_AFS_JOB_SUPPORT** : When this parameter is set to `Y|y`, LSF assumes the user's job is running in an AFS environment, and calls **aklog -setpag** to create a new PAG for the user's job if it is a sequential job, or to create a separate PAG for each task res if the job is a **blaunch** job. LSF then runs the **erenew** script after the TGT is renewed. This script is primarily used to run **aklog**. Finally, LSF assumes that **JOB_SPOOL_DIR** resides in the AFS volume. It kerberizes the child **sbatchd** to get the AFS token so the child **sbatchd** can access **JOB_SPOOL_DIR**

  ```
  LSB_AFS_JOB_SUPPORT=Y|y|N|n
  ```

- **LSB_EAUTH_DATA_REUSE**: When set to `Y`, **blaunch** caches authentication data returned by **eauth -c** when connecting to the first remote execution server in memory. **blaunch** uses this cached data to authenticate subsequent first remote execution servers. If set to `N`, **blaunch** does not cache authentication data. Every time **blaunch** connects to a different authentication, it calls **eauth -c** to fetch new authentication data.

  ```
  LSB_EAUTH_DATA_REUSE=Y|y|N|n
  ```

- **LSB_BSUB_ERR_RETRY**: In some cases, jobs can benefit from being automatically retried in the case of failing for a particular error. When specified, **LSB_BSUB_ERR_RETRY** automatically retries jobs that exit with a particular reason, up to the number of times specified by **RETRY_CNT**.

  ```
  LSB_BSUB_ERR_RETRY=RETRY_CNT[number] ERR_TYPE[error1 error2 ...]
  ```

- **LSF_RES_ALIVE_TIMEOUT**: Controls how long the task res on non-first execution hosts waits (in seconds) before cleaning up the job. If set to 0, this parameter is disabled.

  ```
  SF_RES_ALIVE_TIMEOUT=time_seconds
  ```

- **LSF_DJOB_TASK_REG_WAIT_TIME**: Allows users/admin to define a fixed timeout value to override the internal timeout set by LSF in order to avoid task registration timeout for a large parallel job. Can be configured in `lsf.conf` or set as an environment variable of **bsub**. The environment variable will overwrite the `lsf.conf` configuration. If neither is present, jobRES will use the default value. When it is specified by the environment variable or configured in `lsf.conf`, the value will be directly used by LSF without any adjusting.

  `LSF_DJOB_TASK_REG_WAIT_TIME=time_minutes`
- **LSB_FANOUT_TIMEOUT_PER_LAYER**: Controls how long **sbatchd** waits until the next **sbatchd** replies. Can also be set as an environment variable.

  `LSB_FANOUT_TIMEOUT_PER_LAYER=time_seconds`
- **LSB_DEBUG_EBROKERD**: Sets the debugging log class for the new daemon **ebrokerd**. Only messages belonging to the specified log class are recorded. Used in combination with **LSF_LOG_MASK** which sets the log level.

  `LSB_DEBUG_EBROKERD="log_class [log_class...]"`
- **LSF_DEFAULT_FREQUENCY**: Sets the default CPU frequency for compute nodes when nodes start and when node has finished a job that uses a different CPU frequency. Value is a positive float number with units (GHz or MHz). If no units are set, the default is GHz. If nothing is set for this parameter, the host's nominal CPU frequency will be used.

  `LSF_DEFAULT_FREQUENCY=[float_number][unit]`
- **LSF_MANAGE_FREQUENCY**: Uses a keyword value (N, CORE, or HOST) to set whether the CPU frequency is set for the core (CPU) or by host (node). If the value CORE is set, jobs will require affinity resource requirements. The default value for this parameter is N (not set).

  `LSF_MANAGE_FREQUENCY=N | CORE | HOST`
- **LSF_COLLECT_ENERGY_USAGE**: Determines if the collection of job and node energy usage is enabled on the LSF cluster. This is used for CPU frequency management and energy usage reporting. The default value is N.

  `LSF_COLLECT_ENERGY_USAGE=Y | N`

New and changed accounting and job event fields

The following job event fields are added or changed for LSF 9.1.2.

lsb.events

- **HOST_POWER_STATUS**: LSF logs this event when a host power status is changed, whether by power policy, job, or by the command **badmin hpower**. The **HOST_POWER_STATUS** event is logged to reflect the power status changes.
- **JOB_PROV_HOST**: When a job has been dispatched to a power saved host (or hosts), it will trigger a power state change for the host and the job will be in the PROV state. This event logs those PROV cases.
- cpu_frequency was added to JOB_START, JOB_NEW, and JOB_MODIFY2 to show the CPU frequency at which a job runs.

lsb.acct

- cpu_frequency was added to JOB_FINISH to show the CPU frequency at which a job ran.

Documentation changes

This section summarizes major changes and corrections to the LSF documentation since the release of LSF 9.1.1.

- *Administering IBM Platform LSF* now contains content from “Using Platform LSF HPC Features” guide which is no longer published. Most of the features that are technically supported and relevant to LSF 9.1.x users can be found described in the chapter, "Job Execution and Interactive Jobs".
- Updates to the *IBM Platform LSF Command Reference* and *IBM Platform LSF Configuration Reference* have been made since the released build of the man pages. Please consult these guides for more information on the following:
  - LSF_AFS_BIN_DIR
  - LSB_AFS_JOB_SUPPORT
  - LSB_EAUTH_DATA_REUSE
  - LSF_RES_ALIVE_TIMEOUT
  - LSB_FANOUT_TIMEOUT_PER_LAYER
  - DEFAULT_JOB_OUTDIR
  - HOST_PRE_PROC and HOST_POST_POC
  - JOB_POSTPROC_TIMEOUT and JOB_POSTPROC_TIMEOUT
  - bsub -freq

Known issues

- Host based pre- and post- execution processing timeout algorithm does not consider the program execution time on each execution host in the allocation, which could be too short to abort the processing. LSF admin should configure **JOB_PREPROC_TIMEOUT** to a value to indicate the maximum program runtime expected, the timeout algorithm will consider it as a factor.

  If the job’s host based pre-execution fails, its host based post-execution will be started immediately. It may end up both programs running concurrently if the job is re-dispatched to the same hosts. Configuring **JOB_INCLUDE_POSTPROC** can avoid the situation.
- **brequeue** does not transfer new TGTs to **mbatchd**. If a job is re-queued by the **brequeue** command, the TGT job used is the one cached by **mbatchd**.
- Library requirement for OpenAFS integration permitting jobs with valid Kerberos credentials to access AFS shared directories:

  libkopenafs.so must be provided in one of the following locations: /lib, /lib64, /usr/lib, /usr/lib64, /usr/local/lib, /usr/local/lib64
- CPU and memory affinity scheduling has the following limitations.
  - When reservation is enabled, affinity reservation allocations appear as part of the allocated resources in **bhosts -aff**

    Jobs that are submitted with a `membind=localprefer` binding policy may overcommit the memory of the NUMA node they are allocated to

    bhosts -aff output may occasionally show the total allocated memory on the NUMA nodes of a host as exceeding the maximum memory of the host, this is because the *reservations* that show in **bhosts -aff** overcommit the NUMA node. However, LSF will never allow the allocation of *running* jobs on a host to exceed the maximum memory of a host.
  - When reservation is enabled, and an affinity job requests enough resources to consume an entire node in the host topology. (for example, enough cores to consume an entire socket), LSF will not reserve the socket for the job if there are any jobs running on its cores. In a situation when there are always smaller jobs running consuming cores, then larger jobs that require entire sockets will not be able to reserve resources. The workaround is to require that all jobs have estimated runtimes, and to use time-based reservation.

- LSF does not check the contents or exit code of the erenew script. If erenew contains the wrong command, AFS tokens may not be renewed and LSF will not report any error in the log file. Therefore, users must ensure that the commands in erenew can renew AFS tokens successfully.
- **bmod** cannot change the memory requirement for a running job if a MEM general resource limit is defined for the user in `lsb.resources`.
- Application checkpointing is not supported on 64-bit Windows 7.
- LSF 8.3 **blimits** does not work with 9.1.x binaries.
- For GSLA, a job may pend or receive fewer slots than expected when you ask for a range of slots.

Limitations

- Parallel restart cannot be used if the **mbatchd** is configured to use duplicate event logging (LSB_LOCALDIR is configured in `lsf.conf`).
- Processor number is not detected correctly on POWER7 Linux machines
- NUMA topology may be incorrect after bringing cores offline.
- Cannot remove the energy tag from a job. The workaround is to kill the current job, and submit a new one.

Bugs fixed

The December 2013 release (LSF 9.1.2) contains all bugs fixed before 8 October 2013. Bugs fixed between 25 June 2013 and 8 October 2013 are listed in the document *Fixed Bugs for Platform LSF 9.1.2*. This document is available on Platform LSF's IBM Service Management Connect at http://www.ibm.com/developerworks/servicemanagement/tc/.

Chapter 2. Platform LSF product packages

The Platform LSF product consists of the following packages and files:

- Product distribution packages, available for the following operating systems:

Operating system	Product package
IBM AIX 6 and 7 on IBM Power 6 and 7	lsf9.1.2_aix-64.tar.Z
HP UX B.11.31 on PA-RISC	lsf9.1.2_hppa11i-64.tar.Z
HP UX B.11.31 on IA64	lsf9.1.2_hpuxia64.tar.Z
Solaris 10 and 11 on Sparc	lsf9.1.2_sparc-sol10-64.tar.Z
Solaris 10 and 11 on x86-64	lsf9.1.2_x86-64-sol10.tar.Z
Linux on x86-64 Kernel 2.6 and 3.*x*	lsf9.1.2_linux2.6-glibc2.3-x86_64.tar.Z
Linux on IBM Power 6 and 7 Kernel 2.6 and 3.*x*	lsf9.1.2_linux2.6-glibc2.3-ppc64.tar.Z
Windows 2003/2008/2012/XP/7/8 32-bit	lsf9.1.2_win32.msi
Windows 2003/2008/2012/XP/7/8 64-bit	lsf9.1.2_win-x64.msi
Apple Mac OS 10.x	lsf9.1.2_macosx.tar.Z
zLinux Kernal 2.6, glibc2.4 SLES 10	lsf9.1.2_lnx26-lib24-s390x-64.tar.Z
Cray Linux XE6, XT6, XC-30	lsf9.1.2_lnx26-lib23-x64-cray.tar.Z
ARMv7 Kernel 3.6 glibc 2.15	lsf9.1.2_linux3.6-glibc2.15-armv7.tar.Z

- Installer packages:
  - `lsf9.1.2_lsfinstall.tar.Z`

    This is the standard installer package. Use this package in a heterogeneous cluster with a mix of systems other than x86-64 (except zLinux). Requires approximately 1 GB free space.
  - `lsf9.1.2_lsfinstall_linux_x86_64.tar.Z`

    Use this smaller installer package in a homogeneous x86-64 cluster. If you add other non x86-64 hosts you must use the standard installer package. Requires approximately 100 MB free space.
  - `lsf9.1.2_no_jre_lsfinstall.tar.Z` For all platforms not requiring the JRE. JRE version 1.4 or higher must already be installed on the system. Requires approximately 1 MB free space.
  - `lsf9.1.2_lsfinstall_s390x-64.tar.z` Installer package for zLinux platform. Includes zLinux specific JRE. Requires approximately 300 MB free space.

  The same installer packages are used for LSF Express Edition, LSF Standard Edition, and LSF Advanced Edition.
- Entitlement configuration files:
  - LSF Standard Edition: `platform_lsf_std_entitlement.dat`
  - LSF Express Edition: `platform_lsf_exp_entitlement.dat`.
  - LSF Advanced Edition: `platform_lsf_adv_entitlement.dat`.
- Documentation packages:
  - `lsf9.1.2_documentation.tar.Z`

– `lsf9.1.2_documentation.zip`

Downloading the Platform LSF product packages

Download the LSF installer package, product distribution packages, and documentation packages from IBM Passport Advantage:

www.ibm.com/software/howtobuy/passportadvantage.

Notices

This information was developed for products and services offered in the U.S.A.

IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web

sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Intellectual Property Law
Mail Station P300
2455 South Road,
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application

programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

LSF®, Platform, and Platform Computing are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software

Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Printed in USA

GI13-3413-03