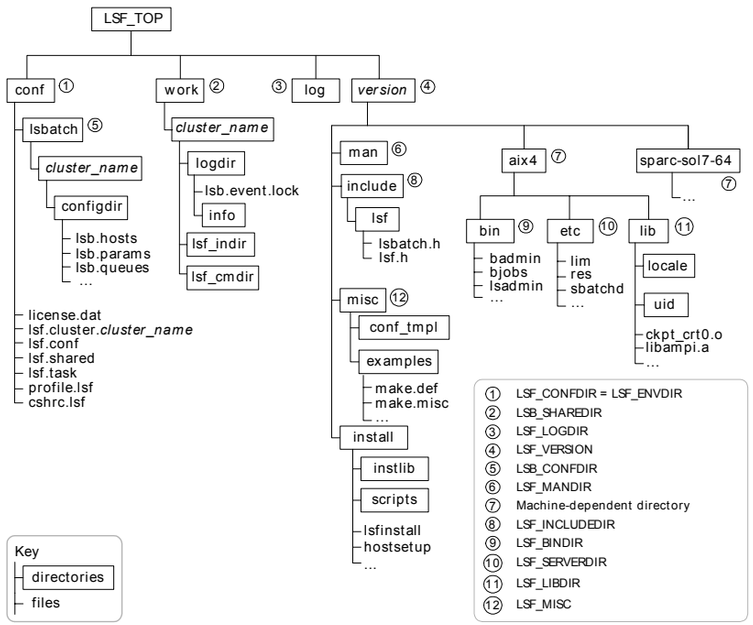
Platform LSF® Quick Reference

Version 5.1

Sample UNIX installation directories



Daemon error log files

Daemon error log files are stored in the directory defined by LSF_LOGDIR in lsf.conf.

LSF base system daemon log files	LSF batch system daemon log files
lim.log.hostname	mbatchd.log.hostname
res.log.hostname	sbatchd.log.hostname
pim.log.hostname	mbschd.log.hostname

Configuration files

lsf.conf, lsf.shared, and lsf.cluster.*cluster_name* are located in LSF_CONFDIR.

lsb.params, lsb.queues, lsb.modules, and lsb.resources are located in LSB_CONFDIR/*cluster_name*/configdir/.

File	Description
lsf.conf	Generic environment configuration file describing the configuration and operation of the cluster
lsf.shared	Definition file shared by all clusters. Used to define cluster name, host types, host models and site-defined resources
lsf.cluster.*cluster_name*	Cluster configuration files used to define hosts, administrators, and locality of site-defined shared resources
lsb.params	LSF batch tunable parameters
lsb.queues	Batch queue configuration file
lsb.modules	Configures LSF scheduler and resource broker plugin modules
lsb.resources	Configures resource allocation limits, exports, and resource usage limits.

Cluster configuration parameters

Variable	Description	UNIX Default
LSF_TOP	Top-level LSF installation directory, must be accessible from all hosts in the cluster	/usr/local/lsf
LSF_BINDIR	Directory containing LSF user commands, shared by all hosts of the same type	LSF_TOP/*version*/*platform*/bin
LSF_CONFDIR	Directory for all LSF configuration files	LSF_TOP/conf
LSF_ENVDIR	Directory containing the lsf.conf file, must be owned by root	/etc
LSF_INCLUDEDIR	Directory containing LSF API header files lsf.h and lsbatch.h	LSF_TOP/*version*/include
LSF_LIBDIR	LSF libraries, shared by all hosts of the same type	LSF_TOP/*version*/*platform*/lib
LSF_LOGDIR	(Optional) Directory for LSF daemon logs, must be owned by root	/tmp
LSF_LOG_MASK	Specifies the logging level of error messages from LSF commands	LOG_WARNING
LSF_MANDIR	Directory containing LSF man pages	LSF_TOP/*version*/man
LSF_MISC	Help files for the LSF GUI tools, sample C programs and shell scripts, and a template for an external LIM (elim)	LSF_TOP/*version*/misc
LSF_SERVERDIR	Directory for all server binaries and shell scripts, and external executables invoked by LSF daemons, must be owned by root, and shared by all hosts of the same type	LSF_TOP/*version*/*platform*/etc
LSB_CONFDIR	Directory for LSF Batch configuration directories, containing user and host lists, operation parameters, and batch queues	LSF_CONFDIR/lsbatch
LSB_SHAREDIR	Directory for LSF Batch job history and accounting log files for each cluster, must be owned by primary LSF administrator	LSF_TOP/work
LSF_LIM_PORT	TCP service port used for communication with the Load Information Manager (LIM) daemons	6879
LSF_RES_PORT	TCP service port used for communication with the Remote Execution Server (RES) daemons	6878
LSB_MBD_PORT	TCP service port used for communication with the Master Batch Daemon (MBD)	6881
LSB_SBD_PORT	TCP service port used for communication with the Slave Batch Daemon (SBD)	6882

Administration commands

Only LSF administrators or root can use these commands.

Command	Description
lsadmin	LSF administrative tool to control the operation of the LIM and RES daemons in an LSF cluster. Enter lsadmin help for a list of subcommands.
lsfinstall	Install LSF using install.config input file
lsfrestart	Restart the LSF daemons on all hosts in the local cluster
lsfshutdown	Shut down the LSF daemons on all hosts in the local cluster
lsfstartup	Start the LSF daemons on all hosts in the local cluster
badmin	LSF administrative tool to control the operation of the LSF Batch system including sbatchd, mbatchd, hosts and queues. Enter badmin help for a list of subcommands.
brun	Forces LSF to run a submitted, pending job immediately on a specified host
brsvadd	By default, adds an advance reservation
brsvdel	By default, deletes an advance reservation

Daemons

Executable Name	Description
lim	Load Information Manager (LIM)—collects load and resource information about all server hosts in the cluster and provides host selection services to applications through LSLIB. LIM maintains information on static system resources and dynamic load indices.
mbatchd	Master Batch Daemon (MBD)—accepts and holds all batch jobs. MBD periodically checks load indices on all server hosts by contacting the Master LIM.
mbschd	Master Batch Scheduler Daemon—performs the scheduling functions of LSF and sends job scheduling decisions to MBD for dispatch. Runs on the LSF master server host.
sbatchd	Slave Batch Daemon (SBD)—accepts job execution requests from MBD, and monitors the progress of jobs. Controls job execution, enforces batch policies, reports job status to MBD, and launches MBD.
pim	Process Information Manager (PIM)—monitors resources used by submitted jobs while they are running. This information is used to enforce resource limits and load thresholds as well as for fairshare scheduling.
res	Remote Execution Server (RES)—accepts remote execution requests from all load sharing applications and handles I/O on the remote host for load sharing processes

User commands

Viewing information about your cluster

Command	Description
bhosts	Displays hosts and their static and dynamic resources
bhpart	Displays information about host partitions
bmgroup	Displays information about host groups
bparams	Displays information about tunable batch system parameters
bqueues	Displays information about batch queues
brsvs	Displays advance reservations
bugroup	Displays information about user groups
busers	Displays information about users and user groups
lshosts	Displays hosts and their static resource information
lsid	Displays the current LSF version number, cluster name and the master host name
lsinfo	Displays load sharing configuration information
lsload	Displays dynamic load indices for hosts

Monitoring jobs and tasks

Command	Description
bhist	Displays historical information about jobs
bjobs	Displays information about jobs
bpeek	Displays stdout and stderr of unfinished jobs
bstatus	Reads or sets external job status messages and data files
lseligible	Displays whether a task is eligible for remote execution

Submitting and controlling jobs

Command	Description
bbot	Moves a pending job relative to the last job in the queue
bchkpnt	Checkpoints a checkpointable job
bkill	Sends a signal to a job
bmig	Migrates a checkpointable or rerunnable job
bmod	Modifies job submission options
bpost	Sends a messages and attaches data files to a job
bread	Reads messages and attached data files from a job
brequeue	Kills and requeues a job
brestart	Restarts a checkpointed job
bresume	Resumes a suspended job
bstop	Suspends a job
bsub	Submits a job
bswitch	Moves unfinished jobs from one queue to another
btop	Moves a pending job relative to the first job in the queue
lsgrun	Executes a task on a set of hosts
lsrun	Runs an interactive task through LSF Base

bsub command

Syntax

bsub [*options*] *command* [*arguments*]

Options

Option	Description
-B	Sends email to you when the job is dispatched and begins execution
-H	Holds the job in the PSUSP state when the job is submitted
-I | -Ip | -Is	Submits a batch interactive job. Blocking. -Ip creates a pseudo-terminal. -Is creates a pseudo-terminal with shell mode support.
-K	Submits a job and waits for the job to finish
-N	Emails the job report to you when the job finishes
-r	Rerun
-x	Exclusive execution
-b *begin_time*	Dispatches the job on or after the specified date and time in the form [[*month*:]*day*:]*hour*:*minute*
-C *core_limit*	Sets a per-process (soft) core file size limit (KB) for all the processes that belong to this job
-c *cpu_time*[/*host_name* | /*host_model*]	Limits the total CPU time the job can use. CPU time is in the form [[*month*:]*day*:]*hour*:*minute*
-D *data_limit*	Per-process (soft) data segment size limit (KB) for each process that belong to the job
-e *error_file*	Stores the standard error output of the job in the specified error file
-E "*pre_exec_command* [*arguments* ...]"	Runs the specified pre-exec command on the execution host before running the job
-f "*local_file op* [*remote_file*]" ...	Copies a file between the local (submission) host and the remote (execution) host. *op* is one of >, <, <<, ><, <>
-F *file_limit*	Per-process (soft) file size limit (KB) for each process that belong to the job
-G *user_group*	Associates job with a specified user group
-i *input_file* | -is *input_file*	Gets the standard input for the job from specified file
-J "*job_name*[*index_list*] %*job_slot_limit*"	Assigns the specified name to the job. For job arrays, specifies the indices of the job array and the maximum number of jobs that can run at any given time. *Index_list* has the form *start*[-*end*[:*step*]], where *start*, *end*, and *step* are positive integers.
-k "*chkpnt_dir* [*chkpnt_period*] [method=*method_name*]"	Makes a job checkpointable and specifies the checkpoint directory, checkpoint period in minutes, and checkpoint method
-L *login_shell*	Initializes the execution environment using the specified login shell
-m "*host_name*[+[*pref_level*]] | *host_group*[+[*pref_level*]] ..."	Runs job on one of the specified hosts. Use plus (+) after the names of hosts or host groups to indicate a preference. Optionally, use a positive integer to indicate a preference level. Higher numbers indicate greater preferences for those hosts. Specify the keyword others to refer to other hosts not listed.
-M *mem_limit*	Specify the memory limit (KB)
-n *min_proc*[,*max_proc*]	Specifies the minimum and maximum numbers of processors required for a parallel job
-o *output_file*	Appends the standard output of the job to the specified file path
-P *project_name*	Assigns job to specified project
-p *process_limit*	Sets the limit of the number of processes to process_limit for the whole job
-q "*queue_name* ..."	Submits job to specified queues
-R "*res_req*"	Specifies host resource requirements
-sp *priority*	Specifies user-assigned job priority which allow users to order their jobs in a queue
-S *stack_limit*	Sets a per-process (soft) stack segment size limit (KB) for each of the processes that belong to the job
-t *term_time*	Specifies the job termination deadline in the form [[*month*:]*day*:]*hour*:*minute*
-U *reservation_ID*	Use advance reservation created with brsvadd
-u *mail_user*	Sends mail to the specified email address
-v *swap_limit*	Set the total process virtual memory limit to swap_limit in KB for the whole job
-w '*dependency_expression*'	LSF will not place your job unless the dependency expression evaluates to TRUE.
-W *run_time*[/*host_name* | /*host_model*]	Sets the run time limit of the job. Run time is in the form [[*month*:]*day*:]*hour*:*minute*.
-Zs	Spools command file to the directory specified by the JOB_SPOOL_DIR in lsb.params, and uses spooled file as the command file for the job
-h	Prints command usage to stderr and exits
-V	Prints LSF release version to stderr and exits

Platform™

Last Update: December 4 2002

www.platform.com
support@platform.com
training@platform.com
doc@platform.com
+1 87PLATFORM (+1 877 528 3676)