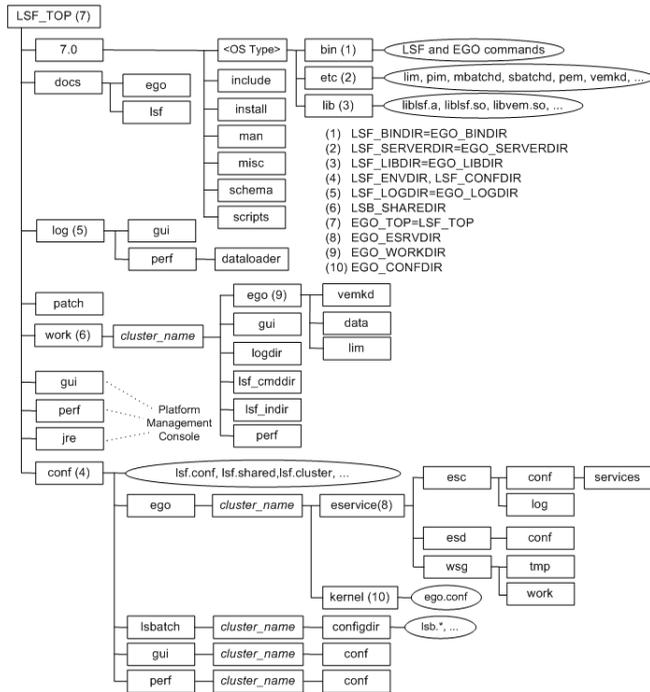## Sample UNIX installation directories

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
pim.log.*host_name*	mbatchd.log.*host_name*
res.log.*host_name*	sbatchd.log.*host_name*
lim.log.*host_name*	mbschd.log.*host_name*

If EGO_LOGDIR is defined in ego.conf, The file lim.log.*host_name* is stored in the directory defined by EGO_LOGDIR.

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
install.config	Options for Platform LSF installation and configuration
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
lsf.licensescheduler	Configures Platform LSF License Scheduler
lsb.applications	Defines application profiles to define common parameters for the same types of jobs
lsb.params	Configures LSF batch parameters

File	Description
lsb.queues	Batch queue configuration file
lsb.resources	Configures resource allocation limits, exports, and resource usage limits
lsb.serviceclasses	Defines service-level agreements (SLAs) in an LSF cluster as service classes, which define the properties of the SLA
lsb.users	Configures user groups, hierarchical fairshare for users and user groups, and job slot limits for users and user groups

Cluster configuration parameters (lsf.conf)

Variable	Description	UNIX Default
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 (if LSF_CONFDIR is not defined)
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
LSF_TOP	Top-level installation directory. The path to LSF_TOP must be shared and accessible to all hosts in the cluster. It cannot be the root directory (/).	Not defined Required for installation
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 lim	7879
LSF_RES_PORT	TCP service port used for communication with res	6878
LSB_MBD_PORT	TCP service port used for communication with mbatchd	6881
LSB_SBD_PORT	TCP service port used for communication with sbatchd	6882

Platform LSF™ Quick Reference

Version 7 Update 6

Administration and accounting commands

Only LSF administrators or root can use these commands.

Command	Description
lsacct	Displays accounting statistics on finished RES tasks in the LSF system
lsadmin	LSF administrative tool to control the operation of the LIM and RES daemons in an LSF cluster, lsadmin help shows all 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
bacct	Reports accounting statistics on completed LSF jobs
badmin	LSF administrative tool to control the operation of the LSF Batch system including sbatchd, mbatchd, hosts and queues, badmin help shows all subcommands
bladmin	reconfigures the Platform LSF License Scheduler daemon (bld)

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. PIM is used to enforce resource limits and load thresholds, and 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
bmgroup	Displays information about host groups and compute units
blimits	Displays information about resource allocation limits of running jobs
bparams	Displays information about tunable batch system parameters
bqueues	Displays information about batch queues
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 master host name
lsinfo	Displays load sharing configuration information
lsload	Displays dynamic load indices for hosts

Monitoring jobs and tasks

Command	Description
bapp	Displays information about jobs attached to application profiles
bhist	Displays historical information about jobs
bjdepinfo	Displays dependency information about jobs.
bjgroup	Displays information about job groups
bjobs	Displays information about jobs
bpeek	Displays stdout and stderr of unfinished jobs
bsla	Displays information about service class configuration for goal-oriented service-level agreement (SLA) scheduling
bstatus	Reads or sets external job status messages and data files

Submitting and controlling jobs

Command	Description
bbot	Moves a pending job relative to the last job in the queue
bchkpnt	Checkpoints a checkpointable job
bgadd	Creates job groups
bgdel	Deletes job groups
bkill	Sends a signal to a job
bmig	Migrates a checkpointable or rerunnable job
bmod	Modifies job submission options
brequeue	Kills and requeues a job
bresize	Releases slots and cancels pending job resize allocation requests.
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

bsub command

Syntax

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

Options

Option	Description
-ar	Specifies the job is autoresizable
-B	Sends email when the job is dispatched
-H	Holds the job in the PSUSP state at submission
-I | -Ip | -Is	Submits a batch interactive job. -Ip creates a pseudo-terminal. -Is creates a pseudo-terminal in shell mode.
-K	Submits a job and waits for the job to finish
-N	Emails the job report when the job finishes
-r	Makes a job rerunnable
-ul	Passes operating system user shell limits to the execution host (UNIX and Linux only)
-x	Exclusive execution
-app *application_profile_name*	Submits the job to the specified application profile
-b *begin_time*	Dispatches the job on or after the specified date and time in the form [[*month*:]*day*:]:*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 [*hour*:]*minute*
-cwd "*current_working_directory*"	Specifies the current working directory for the job.
-D *data_limit*	Sets per-process (soft) data segment size limit (KB) for each process that belongs to the job
-E "*pre_exec_command* [*arguments* ...]"	Runs the specified pre-exec command on the execution host before running the job
-Ep "*post_exec_command* [*arguments* ...]"	Runs the specified post-exec command on the execution host after the job finishes
-e *error_file*	Appends the standard error output to a file
-eo *error_file*	Overwrites the standard error output of the job to the specified file
-F *file_limit*	Sets per-process (soft) file size limit (KB) for each process that belongs to the job
-f "*local_file op* [*remote_file*]" ...	Copies a file between the local (submission) host and remote (execution) host. *op* is one of >, <, <<, ><, <>
-G *user_group*	Associates job with a specified user group
-g *job_group_name*	Associates job with a specified job 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. Job array *Index_list* has the form *start*[-*end*[:*step*]], and %*job_slot_limit is* the maximum number of jobs that can run at any given time.
-k "*chkpnt_dir* [*chkpnt_period*] [method=*method_name*]"	Makes a job checkpointable and specifies the checkpoint directory, period in minutes, and method
-L *login_shell*	Initializes the execution environment using the specified login shell
-M *mem_limit*	Sets the per-process (soft) memory limit (KB)

Option	Description
-m "*host_name* [@*cluster_name*][[!] | +[*pref_level*]] | *host_group*[[!] |+[*pref_level*]] | *compute_unit*[[!] |+[*pref_level*]]..."	Runs job on one of the specified hosts. Plus (+) after the names of a host or group indicates a preference. Optionally, a positive integer indicates a preference level with higher numbers indicating a greater preference.
-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 to a file
-oo *output_file*	Overwrites the standard output of the job to the specified file
-P *project_name*	Assigns job to specified project
-p *process_limit*	Limit the number of processes for the whole job
-Q "[*exit_code* …] [EXCLUDE(*exit_code* …)]"	Specifies automatic job requeue exit values and exclusive job requeue exit values
-q "*queue_name* ..."	Submits job to one of the specified queues
-R "*res_req*" [-R "*res_req*" …]	Specifies host resource requirements
-rnc *resize_notification_cmd*	Specifies full path of an executable invoked on the first execution host when the job allocation is modified
-S *stack_limit*	Sets a per-process (soft) stack segment size limit (KB) for each process that belongs to the job
-s *signal*	Send *signal* when a queue-level run window closes
-sla *service_class_name*	Specifies the service class where the job is to run
-sp *priority*	Specifies user-assigned job priority allowing users to order their jobs in a queue
-T *thread_limit*	Sets the limit of the number of concurrent threads for the whole 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 (KB) for the whole job
-W *run_time*[/*host_name* | /*host_model*]	Sets the run time limit of the job in the form [*hour*:]*minute*
-w '*dependency_expression*'	Places a job when the dependency expression evaluates to TRUE
-wa '*signal*'	Specifies the job action to be taken before a job control action occurs
-wt '[*hour*:]*minute*'	Specifies the amount of time before a job control action occurs that a job warning action is to be taken
-Zs	Spools a command file for the job to the directory specified by the JOB_SPOOL_DIR in lsb.params
-h	Prints command usage to stderr and exits
-V	Prints LSF release version to stderr and exits