License server configuration file

The information on this page refers to LM-X v5.5 or newer, which extended the LICENSE_FILE setting to support specifying a file path in addition to specifying a file name, and changed the LICENSE_FILE setting name to LICENSE_PATH. If you are using a previous version of LM-X, please see the documentation for the previous version.

The vendor may supply a license server configuration file, named lmx-serv.cfg by default. This configuration file is an ASCII text file, which can be opened and modified using any text editor. You may replace the existing information in the configuration file as needed.

For example, assume we are using a floating license. This type of license allows any number of users to have the software installed, but only a certain number of users to use the software simultaneously. When all allotted licenses are in use, other users must wait until a license becomes available to use the software. If you want to use software licenses most efficiently and implement fair/desired distribution of licenses, then as an administrator of the LM-X server, you may set the following options in the LM-X License Server configuration file:

- Permissions. Setting permissions lets you allow or deny individual users/groups use of the license server according to your organization's specific needs. Permissions can be based on a set of rules that include permissions for normal checkouts as well as license borrowing.
- Reservations. You can reserve a specified number of licenses that can be used by individual users or groups. Reservations can also be done using a set of rules, allowing you to specify the reservation order. Some users or groups can be given higher priority than others.
- Limitations. You can limit the number of licenses that can be used by individual users or groups. Limitations can be done by a set of rules. In particular, limiting of users is done by a first match rule, so if a user belongs to more than one group specified in restrictions, the first restriction will apply to that user.

The configuration file includes instructions for using each setting in the file, which may include the following, depending on the options provided by your vendor. Some of the configuration settings can also be specified using the web-based UI, as described in Administration.

Syntax	Description	Examples
TCP_LISTEN_PORT = *port number*	The TCP port number the license server will listen on. • TCP port is used for data traffic protocol. The default TCP port is 6200. • UDP port is used for automatic server discovery protocol. The UDP port is fixed to 6200 and cannot be changed. See http://www.iana.org/assignments/port-numbers for more information.	TCP_LISTEN_PORT = 6200
TCP_BIND_ADDRESS = *IP_address_1 IP_address_2*	Limit which networks the license server allows for client connections. When this setting is specified, the license server will only accept clients that connect from a network that uses the specified IP addresses. You can specify only one address for each IP version (one for IPV4 and one for IPV6). This setting is useful when the license server is connected to more than one network (has more than one IP address) and you want to limit allowed connections based on which network the client is on. When this setting is unspecified, the license server accepts clients from all available networks.	TCP_BIND_ADDRESS = 192.168.21.321 8000:8000:8000:8000:abcd:1234:12df:fd54

HAL_SERVER*server_number* = [*port*]@*hostname* or HAL_SERVER*server_number* = [*port*]@*IP_address* **Note**: Port is optional.	High Availability Licensing (HAL) servers, which enable redundant servers, so if one server goes down, two others will still work. HAL consists of 3 specified servers, at least 2 of which must be up and running at all times. Each HAL_SERVER line indicates a license server that has HAL enabled by its license(s). Each HAL server has a specific role, and should be specified in terms of how many resources each server has: • HAL_SERVER1 is your master server, which allows both CHECKOUT and BORROW. HAL_SERVER1 should be your most powerful server. • HAL_SERVER2 is your first slave server, which allows CHECKOUT but denies BORROW in the event that your master server goes down. HAL_SERVER2 should be your second most powerful server. • HAL_SERVER3 is part of your configuration to ensure that everything works as expected, and does not allow any CHECKOUT or BORROW requests. HAL_SERVER3 should be your least powerful server. **Important**: The HAL_SERVER list must be identical on all your servers for HAL to function properly	HAL_SERVER1 = 6200@server1 HAL_SERVER2 = 6200@server2 HAL_SERVER3 = 6200@server3
LOG_FILE = *path*	The log file path. Specifying the full path is preferred. If you do not specify this setting, the default is used: On Windows the default is lmx-serv .log, under the license server directory. On Unix, the default location for the log file is in the directory from which the license server was started	LOG_FILE = c:\program files\lmx-server.log LOG_FILE = /home/user1/lmx-serv.log
LOG_FORMAT = NORMAL or EXTENDED	The format for the log file. The default setting for the log file format is NORMAL. Specifying EXTENDED causes additional information to be included in the log file, such as license server HostIDs, whether the license server is a virtual machine, etc. Setting the log file format to EXTENDED is particularly useful for debugging purposes.	LOG_FORMAT= NORMAL LOG_FORMAT=EXTENDED
LOG_EXCLUDE = *message1*, *message2*, etc.	Exclude messages from the log. The following messages can be excluded: CHECKOUT, CHECKIN, STATUS, BORROW, BORROW_RETURN, REMOVE_USER, REMOTE_RESTART or REMOTE_SHUTDOWN.	LOG_EXCLUDE = CHECKOUT, CHECKIN, STATUS
LOGFILE_ROTATE_INTERVAL = *rotation_interval*	The interval for log file rotation. The value may be set to "day" (every day at midnight), "week" (every Monday at midnight), or "month" (the first day of every month at midnight). After rotation, the old log file will be named *filename*.log.*rotation_date* in the format yyyy-mm-dd. A message indicating the location of the rotated log file is added to the end of the old log file and the beginning of the new log file, as follows: "Log file was rotated and saved to *filename*."	LOGFILE_ROTATE_INTERVAL = day
MIN_USER_REMOVE_TIME = *time in seconds*	Minimum time, in seconds, that must elapse from the connection before a user can be removed using lmxendutil. The specified time must be equal to or greater than the number of seconds specified by your application vendor. Default minimum time is 120 seconds. If the time is set to -1, user removals will not be allowed.	MIN_USER_REMOVE_TIME = 120

LICENSE_PATH = *path*	The full or partial path under which to search for license file(s). Specifying the license file name is optional, and you can specify one or multiple paths as needed. The path must be lowercase. On Windows: If no path is set, the license server will look for *vendor*.lic in the same directory as the license server. On Unix: If no path is set, the license server will look for /usr/x-formation/*vendor*.lic.	LICENSE_PATH = c:\ LICENSE_PATH = d:\server LICENSE_PATH = d:\server\network.lic LICENSE_PATH = c:\extra_file.lic LICENSE_PATH = /home/user1 /floating_license.lic LICENSE_PATH = /home/user1 /floating_license2.lic
USAGE_DATABASE = *database path*	Pay-per-use usage database (used for billing purposes). See Pay Per Use feature for EXTENDED information, including database format and an example of data printout.	USAGE_DATABASE = d:\server\usage.db USAGE_DATABASE = /home/user1/usage.db
USAGE_LEVEL= *detail level*	Specify pay-per-use detail level. • STANDARD includes basic usage information. • EXTENDED includes user information in addition to the basic usage information.	USAGE_LEVEL = STANDARD
USAGE_WRITE_INTERVAL= *number of actions*	Specify the number of pay-per-use actions (checkouts, checkins, etc.) after which pay-per-use records will be written to the pay-per-use database file. The default setting is 1000.	USAGE_WRITE_INTERVAL = 1000
REMOTE_ACCESS_PASSWORD = *password*	Remote administration password (used when remotely stopping and restarting the license server and removing users from it). The password is case-sensitive.	REMOTE_ACCESS_PASSWORD = MyPassword123
FAST_QUEUE = *feature1, feature2*, etc. or FAST_QUEUE = ALL	Fast queuing allows requests that can be fulfilled immediately to be fulfilled. For example, if a client is waiting for two licenses, and only one license is immediately available, another client that needs only one license can bypass the queue and take the single license without waiting. Default behavior of license queuing is to put the client at the end of the queue regardless whether the license request could be satisfied.	FAST_QUEUE = f2, d5, app2

ALLOW_IPADDR_ALL = *one or more IP addresses* ALLOW_IPADDR_*feature name = one or more IP addresses* (must be either specific A.B.C.D or with wildcards; e.g., A.B.*) DENY_IPADDR_ALL = *one or more IP addresses* DENY_IPADDR_*feature name = one or more IP addresses* (must be either specific A.B.C.D or with wildcards; e.g., A.B.*) ALLOW_HOST_ALL = *one or more hosts* ALLOW_HOST_*feature name = one or more hosts* DENY_HOST_ALL = *one or more hosts* DENY_HOST_*feature name = one or more hosts* ALLOW_USER_ALL = *one or more users* ALLOW_USER_*feature name = one or more users* DENY_USER_ALL = *one or more users* DENY_USER_*feature name = one or more users* **Note**: For host, you can use a hostname or use "localhost" to specify the current machine. For IP address, you can specify a complete address (A.B.C.D) or use wildcards; e.g., A.B.*).	Allow/deny specific clients from using the license server. The allow/deny rules work as follows: • Rules are attempted to be matched in the order they are written. • If no rule matches the specific client, then that client is allowed.	The following example will deny all clients except that with hostname 'trusted'. This applies to all features. ALLOW_HOST_ALL = trusted DENY_IPADDR_ALL = *.*.*.* The following example will allow clients on only 2 subnets, user Administrator and root from any host and deny everyone else. This applies to all features. ALLOW_IPADDR_ALL = 192.168.1.* 192.168.2.* ALLOW_USER_ALL = Administrator root DENY_IPADDR_ALL = *.*.*.* The following example will deny clients on localhost, deny the machines with hostname 'untrusted' and 'crackerjack', allow clients on the internal network, and deny everyone else. This applies to the feature f2. DENY_HOST_f2 = localhost untrusted crackerjack ALLOW_IPADDR_f2 = 192.168.*.* DENY_IPADDR_f2 = *.*.*.*

ALLOW_BORROW_IPADDR_ALL = *one or more hosts* ALLOW_BORROW_IPADDR_*feature name = one or more hosts* DENY_BORROW_IPADDR_ALL = *one or more hosts* DENY_BORROW_IPADDR_*feature name = one or more hosts* ALLOW_BORROW_HOST_ALL = *one or more hosts* ALLOW_BORROW_HOST_*feature name = one or more hosts* DENY_BORROW_HOST_ALL = *one or more hosts* DENY_BORROW_HOST_*feature name = one or more hosts* ALLOW_BORROW_USER_ALL = *one or more users* ALLOW_BORROW_USER_*feature name = one or more users* DENY_BORROW_USER_ALL = *one or more users* DENY_BORROW_USER_*feature name = one or more users* **Note**: For host, you can use a hostname or use "localhost" to specify the current machine. For IP address, you can specify a complete address (A.B.C.D) or use wildcards; e.g., A.B.C.*).	Allow/deny specific clients from borrowing licenses.	The following example will allow the specific users, and deny host and IP addresses on the list from borrowing any feature. Everyone else will be allowed. ALLOW_BORROW_USER_ALL = daisy harry tom DENY_BORROW_HOST_ALL = server1 machine5 DENY_BORROW_IPADDR_ALL = 192.168.3.* 192.168.4.* The following example will allow the specific users and deny everyone else from borrowing f2. ALLOW_BORROW_USER_f2 = lazyjack rabbit joeuser DENY_BORROW_IPADDR_f2 = *.*.*.*
LIMIT_USER_*feature name___limit count = one or more users* LIMIT_HOST_*feature name___limit count = one or more hosts* LIMIT_IPADDR_*feature name_limit count = one or more hosts* **Note**: For host, you can use a hostname or use "localhost" to specify the current machine. For IP address, you can specify a complete address (A.B.C.D) or use wildcards; e.g., A.B.C.*).	Limit the number of licenses that can be used by individual users or groups to implement fair/desired distribution of licenses. Limiting of users is done by a first match rule, so if a user belongs to more than one group specified in restrictions, the first restriction will apply to that user.	LIMIT_USER_f2_5 = harry joe sam LIMIT_IPADDR_f3_3 = 192.168.2.* 192.168.4.*
RESERVE_USER_*feature name___reserve count = one or more users* RESERVE_HOST_*feature name___reserve count = one or more hosts* RESERVE_IPADDR_*feature name_reserve count = one or more hosts* **Note**: For host, you can use a hostname or use "localhost" to specify the current machine. For IP address, you can specify a complete address (A.B.C.D) or use wildcards; e.g., A.B.C.*).	Reserve a number of licenses that can be used by individual users or groups to implement fair/desired distribution of licenses. Reservation of users is done by a first match rule, so if a user belongs to more than one group specified in the rules, the first rule will apply to that user.	RESERVE_USER_f2_5 = harry joe sam RESERVE_IPADDR_f3_3 = 192.168.2.* 192.168.4.*

BORROW_LIMIT_COUNT_ALL = *limit count* BORROW_LIMIT_COUNT_*feature name = limit count* IGNORE_BORROW_LIMIT_COUNT_USER_*feature name = one or more users* IGNORE_BORROW_LIMIT_COUNT_HOST_*feature name = one or more hosts* IGNORE_BORROW_LIMIT_COUNT_IPADDR_*feature name = one or more IP addresses* **Note***:* You can use IGNORE_BORROW_LIMIT_* flag to whitelist small and specific predicates blacklisted by broader BORROW_LIMIT_* predicate.	Limit/Do not limit the number of licenses that can be borrowed to prevent all licenses from being borrowed at the same time.	BORROW_LIMIT_COUNT_f2 = 1 BORROW_LIMIT_COUNT_ABCDEF = 5 The following example will allow the user ADMIN, and deny host and IP addresses on the list from borrowing more than 100 features. BORROW_LIMIT_COUNT_F1 = 100 IGNORE_BORROW_LIMIT_COUNT_USER_F1 = admin IGNORE_BORROW_LIMIT_COUNT_HOST_F1 = server IGNORE_BORROW_LIMIT_COUNT_IPADDR_F1 = 192.168.1.*
BORROW_LIMIT_HOURS_ALL = *limit hours* BORROW_LIMIT_HOURS_*feature name = limit hours* IGNORE_BORROW_LIMIT_HOURS_USER_*feature name = one or more users* IGNORE_BORROW_LIMIT_HOURS_HOST_*feature name = one or more hosts* IGNORE_BORROW_LIMIT_HOURS_IPADDR_*feature name = one or more IP addresses* **Note**: You can use IGNORE_BORROW_LIMIT_* flag to whitelist small and specific predicates blacklisted by broader BORROW_LIMIT_* predicate.	Limit/Do not limit the number of hours licenses can be borrowed to prevent licenses from being borrowed for too long.	The following example will allow the specific users, and deny host and IP addresses on the list from borrowing any feature. Everyone else will be allowed. BORROW_LIMIT_HOURS_f2 = 1 BORROW_LIMIT_HOURS_ABCDEF = 5 The following example will allow the user admin, and deny host and IP addresses on the list from borrowing more than 100 features. BORROW_LIMIT_HOURS_F1 = 10 IGNORE_BORROW_LIMIT_HOURS_USER_F1 = admin IGNORE_BORROW_LIMIT_HOURS_HOST_F1 = server IGNORE_BORROW_LIMIT_HOURS_IPADDR_F1 = 192.168.1.*

FEATURE *featurename*
{
feature settings
}
	Specify licenses directly within the configuration file to eliminate the need to have both a license file and configuration file for the license server. You can specify any features from one or more license files.
	
	The content must be specified within the *_START_LICENSE_* and *_END_LICENSE_* clauses.
		*_START_LICENSE_*
		FEATURE F1
		{
		VENDOR = XYZ
		}
		*_END_LICENSE_*

GROUP_*name = member1 member2*	Specify a group name and the group members to which you want to apply restrictions, limitations and reservations. Creating groups can make these features easier to use and help you to avoid/remove redundancies from the configuration file. You can create groups that contain users, host names and IP addresses. Groups can contain any other group, and there is no limit on the number of members that can be included in a group. The names of groups and group members are case-insensitive.	The following example creates a group named "hr" with one member, "anna," and another group, "employees," which contains three individual users (joe, mary, and sam), plus includes the group "hr" as a sub-group. GROUP_hr = anna GROUP_employees = joe mary sam hr After creating groups, you can apply permissions, reservations, and limitations described above to those groups in the same way you would individual users; for example: DENY_USER_f1 = employees RESERVE_USER_f2_2 = employees LIMIT_USER_f3_2 = hr

DENIAL_STORE_PERIOD = *time_in_seconds*	Specify how long the license server will keep denial information, in seconds. • Valid values are between one minute and one year, expressed in seconds. • The value is numeric only; do not use commas or periods in the entered value. • The default duration is one day (86400 seconds).	The following example sets the denial storage period to 1 day: DENIAL_STORE_PERIOD = 86400