

Managing the Options File

The options file allows the license administrator to control various operating parameters within the constraints of the license model. Users are identified by their user name, host name, display, IP address, or PROJECT (which is set with the LM_PROJECT environment variable).

For concurrent (floating) licenses, the license administrator can:

- Allow the use of features
- Deny the use of features
- Reserve licenses

The concurrent licenses can be held either in license files or in fulfillment records within trusted storage.

For activatable licenses, the license administrator can:

- Allow activation of licenses in a specific fulfillment record
- Deny activation of licenses in a specific fulfillment record

For all licenses, the license administrator can:

- Restrict the number of licenses available
- Control the amount of information logged about license usage
- Enable a report log file

Options files allow you, as the license administrator, to be as secure or open with licenses as you like.

Lines in the options file are limited to 2048 characters. The \ character is a continuation character in options file lines.

- PROJECT identification (set by LM_PROJECT) in options file was introduced in version 7.0 vendor daemon.
- Option file control for licenses held in fulfillment records in trusted storage has been introduced in 11.3 vendor daemon.

Creating an Options File



Task

To create an options file:

1. Use the appropriate options listed in [Options File Syntax](#) to create the options file for a vendor daemon using any text editor.
2. Locate the options file anywhere; however, it is recommended that the options file be placed in the same directory as the license file.
3. Add the path to the options file in the license file as the fourth field on the VENDOR line for the application's vendor daemon. For example:

```
VENDOR sampled /etc/sampled \  
[options=]/sample_app/sampled/licenses/sampled.opt
```

enables the `sampled` vendor daemon to look at the specified options file.

If the path is omitted, the vendor daemon automatically looks for a file according to the following criteria:

- The name of the file is `vendor.opt`, where *vendor* is the vendor daemon name.
- The directory that contains the license file used by the license server manager.



Note: The default options file name, `vendor.opt`, introduced in version 6 vendor daemon.

Options File Syntax

Below is an overview of the options file syntax. See [Options File Examples](#) for examples and additional information.

Each line of the file controls one option. [Table 13-1](#) lists the option keywords.

Table 13-1: Option Keywords

Option Keyword	Description
BORROW_LOWWATER	Set the number of BORROW licenses that cannot be borrowed.
DEBUGLOG	Writes debug log information for this vendor daemon to the specified file (version 8.0 or later vendor daemon).
EXCLUDE	Deny a user access to a feature.
EXCLUDE_BORROW	Deny a user the ability to borrow BORROW licenses.
EXCLUDE_ENTITLEMENT	Deny a user the ability to activate licenses held in a fulfillment record in trusted storage.

Table 13-1: Option Keywords (cont.)

Option Keyword	Description
EXCLUDEALL	Deny a user access to <i>all</i> features served by this vendor daemon.
FQDN_MATCHING	Sets the level of host name matching.
GROUP	Define a group of users for use with any options.
GROUPCASEINSENSITIVE	Sets case sensitivity for user and host lists specified in GROUP and HOST_GROUP keywords.
HOST_GROUP	Define a group of hosts for use with any options (version 4.0 or later).
INCLUDE	Allow a user to use a feature.
INCLUDE_BORROW	Allow a user to borrow BORROW licenses.
INCLUDE_ENTITLEMENT	Allow a user to activate licenses held in a fulfillment record in trusted storage.
INCLUDEALL	Allow a user to use <i>all</i> features served by this vendor daemon.
LINGER	Allow a user to extend the linger time for a feature beyond its checkin.
MAX	Limit usage for a particular feature/group—prioritizes usage among users.
MAX_BORROW_HOURS	Changes the maximum borrow period for the specified feature.
MAX_OVERDRAFT	Limit overdraft usage to less than the amount specified in the license.
NOLOG	Turn off logging of certain items in the debug log file.
REPORTLOG	Specify that a report log file suitable for use by the FLEXnet Manager license usage reporting tool be written.
RESERVE	Reserve licenses for a user or group of users/hosts.
TIMEOUT	Specify idle timeout for a feature, returning it to the free pool for use by another user.
TIMEOUTALL	Set timeout on all features.

Comments

Include comments in your options file by starting each comment line with a pound sign, #.

Specifying Features

When used within an options file entry, the feature name can be modified with an optional keyword-value pair to fully qualify it. This notation is used for distinguishing a particular group of licenses when there are multiple FEATURE lines for a single feature. The following syntax is used:

feature: keyword=value

For example:

f1:VERSION=2.0

specifies the version 2.0 pool of licenses for feature f1.



Note: A colon (:) is a valid feature name character. If colons are in your feature names, specify a group of licenses with the following alternative syntax using quotation marks and spaces:

"feature keyword=value"

The following option keywords are used as feature name modifiers to denote a specific group of licenses:

- VERSION=
- HOSTID=
- EXPDATE=
- KEY=
- SIGN=
- ISSUER=
- NOTICE=
- VENDOR_STRING= (if configured by the publisher as a pooling component)
- dist_info=
- user_info=
- asset_info=

If the USER_BASED or HOST_BASED keywords appear in a feature line, this feature specification syntax must be used to qualify the feature.

Using a package name in place of a feature name applies the option to all of the components in the package.

Specifying License Restrictions Using Type

Some option keywords restrict who may use licenses or where licenses may be used. These options take a type argument that specifies what the restriction is based on.

When using the option keywords EXCLUDE, EXCLUDE_ENTITLEMENT, EXCLUDEALL, EXCLUDE_BORROW, INCLUDE, INCLUDE_ENTITLEMENT, INCLUDEALL, INCLUDE_BORROW, MAX, and RESERVE, the following values can be used for type:

- **USER**—user name of the user executing the FLEXenabled application. User names are case sensitive and cannot contain spaces.
- **HOST**—system host name or IP address where the application is executing. Host names are case sensitive. The IP address can contain wildcard characters.

The IP-address can contain wildcard characters.

When using the option keywords EXCLUDE, EXCLUDEALL, EXCLUDE_BORROW, INCLUDE, INCLUDEALL, INCLUDE_BORROW, MAX, and RESERVE, the following values can be used for type:

- **DISPLAY**—display where the application is displayed. On UNIX, DISPLAY is `/dev/ttyxx` (which is always `/dev/tty` when an application is run in the background) or the X-Display name. On Windows, it is the system name or, in the case of a terminal server environment, the terminal server client name. Display names are case sensitive.
- **INTERNET**—IP address of the system where the application is executing (wildcard characters can be used in the IP address)
- **PROJECT**—LM_PROJECT environment variable set by the user who is executing the FLEXenabled application. Project names are case sensitive.

On Windows (without terminal server), the HOST and DISPLAY names are both set to the system name. For licenses that allow checkouts from a terminal server (TS_OK keyword in the feature line), the USER, HOST, and DISPLAY names can be different from one another.

The types listed above take a single member. For example:

```
EXCLUDE coolsoft USER joe
```

To specify a list of users or hosts, first define the list using the GROUP or HOST_GROUP option lines, then use the GROUP or HOST_GROUP type to specify the group name. For example:

```
GROUP stars joe barbara susan
EXCLUDE coolsoft GROUP stars
```

- IP address as a HOST specification introduced in version 8 vendor daemon.
- Colons in feature names introduced in version 8 vendor daemon.

BORROW_LOWWATER

This option is used for licenses held in license files. When licenses are available in trusted storage, activation is normally provided instead of BORROW.

```
BORROW_LOWWATER feature[:keyword=value] n
```

Sets the number of licenses for a BORROW feature that cannot be borrowed.

Table 13-2: BORROW_LOWWATER Terms

Term	Description
<i>feature</i>	Name of feature being affected.
<i>keyword=value</i>	Feature name modifier to denote a group of licenses. See Specifying Features for details.
<i>n</i>	Number of licenses that cannot be borrowed via license borrowing.

For example, if a feature “f1” has a count of 10 and borrowing is enabled in the application and on the FEATURE line:

```
FEATURE f1 ... 10 ... BORROW SIGN=...
```

the following line in the options file allows only 7 licenses to be borrowed.

BORROW_LOWWATER f1 3

DEBUGLOG

DEBUGLOG *[+]**debug_log_path*

Specifies a location for the debug log output from the vendor daemon associated with this options file. Preceding the *debug_log_path* with a + character appends logging entries; otherwise, the file is overwritten each time the daemon is started. Note that this affects output from only the vendor daemon associated with this options file. The debug log output of *lmadmin* or *lmgrd* and any other vendor daemons in the same license file is not captured in this file.

On Windows, path names which include spaces have to be enclosed in double quotes. If *lmgrd* is started as a service, the default location for the report log file is the *c:\winnt\System32* folder unless a fully qualified path is specified.

See Also:
[Configuring the License Server Manager as a Windows Service](#)
[lmswitch](#)
[Debug Log File](#)—Debug log output restricted to that of just the vendor daemon introduced in version 8 vendor daemon.

EXCLUDE

This option applies to concurrent licenses held in license files and trusted storage.

EXCLUDE *feature[:keyword=value]* *type {name | group_name}*

Excludes a user or predefined group of users from the list of who is allowed to use the feature. EXCLUDE supersedes INCLUDE; conflicts between the EXCLUDE list and the INCLUDE list are resolved by the EXCLUDE taking precedence.

Table 13-3: EXCLUDE Terms

Term	Description
<i>feature</i>	Name of the feature or package being affected.
<i>keyword=value</i>	Feature name modifier to denote a group of licenses. See “ Specifying Features ” for details.
<i>type</i>	One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See “ Specifying License Restrictions Using Type ” for details.
<i>name</i>	Name of an item of type <i>type</i> for which license usage is excluded.
<i>group_name</i>	Name of the group to exclude.



Task *To exclude the user hank from the list of users able to use feature f1:*

```
EXCLUDE f1 USER hank
```

EXCLUDE_BORROW

This option is used for licenses held in license files. When licenses are available in trusted storage, activation is normally provided instead of BORROW.

```
EXCLUDE_BORROW feature[:keyword=value] type \  
               {name | group_name}
```

Excludes a user or predefined group of users from the list of who is allowed to borrow licenses for this BORROW feature. EXCLUDE_BORROW supersedes INCLUDE_BORROW; conflicts between the EXCLUDE_BORROW list and the INCLUDE_BORROW list are resolved by the EXCLUDE_BORROW taking precedence.

Table 13-4: EXCLUDE_BORROW Terms

Term	Description
<i>feature</i>	Name of the feature being affected.
<i>keyword=value</i>	Feature name modifier to denote a group of licenses. See Specifying Features for details.
<i>type</i>	One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See Specifying License Restrictions Using Type for details.
<i>name</i>	Name of an item of type <i>type</i> for which license borrowing is excluded.
<i>group_name</i>	Name of the group to exclude from borrowing.



Task *To exclude the user fred from the list of users able to borrow feature f1 assuming the feature has the BORROW attribute:*

```
EXCLUDE_BORROW f1 USER fred
```

EXCLUDE_ENTITLEMENT

This option only applies to licenses held in trusted storage and supplied using activation.

```
EXCLUDE_ENTITLEMENT entitlementId type {name | group_name}
```

Excludes a user or pre-defined group of users, etc., from the list of who is allowed to activate the licenses contained in a fulfillment record held in trusted storage. EXCLUDE_ENTITLEMENT supersedes INCLUDE_ENTITLEMENT; conflicts between the EXCLUDE_ENTITLEMENT list and the INCLUDE_ENTITLEMENT list are resolved by the EXCLUDE_ENTITLEMENT taking precedence.

Table 13-5: EXCLUDE_ENTITLEMENT Terms

Term	Description
<i>entitlementId</i>	The entitlement Id used when requesting a license activation.
<i>type</i>	One of USER, HOST, GROUP, or HOST_GROUP. See Specifying License Restrictions Using Type for details.
<i>name</i>	Name of an item of type <i>type</i> for which license usage is excluded.
<i>group_name</i>	Name of the group to exclude.



Important: To exclude the user “pete” from the list of users able to activate licenses provided in the fulfillment record specified by the entitlement ID “AB456”:

```
EXCLUDE_ENTITLEMENT AB456 USER pete
```

EXCLUDEALL

This option applies to concurrent licenses held in license files and trusted storage.

```
EXCLUDEALL type {name | group_name}
```

Excludes a user or predefined group of users from the list of who is allowed to use all features served by this vendor daemon.

Table 13-6: EXCLUDEALL Terms

Term	Description
<i>type</i>	One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See Specifying License Restrictions Using Type for details.
<i>name</i>	Name of an item of type <i>type</i> for which license usage is excluded.
<i>group_name</i>	Name of the group to exclude.

To exclude any user on the system called **chaos** using all features served by this vendor daemon:

EXCLUDEALL HOST chaos

FQDN_MATCHING

This option applies to all licenses held in license files or trusted storage.

FQDN_MATCHING exact | lenient

Sets the level to which host names used in HOST type-specifiers must match the host name sent by the FLEXenabled application. The application is configured to send either its host name or its fully qualified domain name (FQDN) to the vendor daemon for validation with HOST type-specifiers. Check with your software publisher to determine fully qualified domain name support.

Table 13-7: FQDN_MATCHING Terms

Term	Description
exact	The host name in the HOST type specifier must match in content and format to that sent by the application. This is the default setting.
lenient	The host name sent by the application needs match to the extent supplied in the HOST type specifier or by the application, which ever is less restrictive.

Only the last FQDN_MATCHING keyword in the options file has effect; all others are ignored.

Table 13-8 shows the outcome of matching attempts between HOST type-specifiers in the options file and host names sent by the application.

Table 13-8: Host Name Matching Matrix

Options File Settings		Application configured for FQDN—sends myhost.abc.com	Application not configured for FQDN—sends myhost
FQDN_MATCHING exact	INCLUDE feature HOST myhost	no	yes
	INCLUDE feature HOST myhost.abc.com	yes	no
FQDN_MATCHING lenient	INCLUDE feature HOST myhost	yes	yes
	INCLUDE feature HOST myhost.abc.com	yes	yes

Examples

Consider the following example that demonstrates restrictive host name matching:

```
INCLUDE f1 HOST myhost.abc.com
FQDN_MATCHING exact
```

This includes myhost.abc.com on the list of hosts able to use feature f1. Furthermore, the host name sent by the application must be a fully qualified domain name that matches myhost.abc.com exactly.

In contrast, consider this example, which is less restrictive:

```
INCLUDE f2 HOST myhost.abc.com
FQDN_MATCHING lenient
```

This includes myhost.abc.com on the list of hosts able to use feature f2. The license rights are authenticated and a checkout allowed if any of the following match:

- The FQDN - myhost.abc.com
- The host name - myhost
- The domain name - .abc.com

The example below is even more lenient:

```
INCLUDE f2 HOST myhost
FQDN_MATCHING lenient
```

This includes the host name, myhost, on the list of hosts for feature f3. Since lenient matching is specified, host names such as myhost, myhost.abc.com, and myhost.xyz.com match, whereas yourhost or yourhost.abc.com do not match.

See Also

[“Specifying License Restrictions Using Type”](#)
FQDN_MATCHING introduced in version 9.3 client library and vendor daemon.

GROUP

```
GROUP group_name user_list
```

Defines a group of users for use in INCLUDE, INCLUDEALL, INCLUDE_ENTITLEMENT, EXCLUDE, EXCLUDEALL, EXCLUDE_ENTITLEMENT, and RESERVE option lines.

Table 13-9: GROUP Terms

Term	Description
group_name	Name of the group being defined. Group names are case sensitive.
user_list	List of user names in that group. Names are case sensitive and cannot contain spaces. Set the GROUPCASEINSENSITIVE options file keyword to turn on case insensitivity. See GROUPCASEINSENSITIVE .

Multiple GROUP lines for the same group name add all the specified users into the group.

To define the group **Hackers** consisting of **bob**, **howard**, and **james**:

```
GROUP Hackers bob howard james
```



Note: USER_GROUP is an alias for GROUP.

GROUPCASEINSENSITIVE

GROUPCASEINSENSITIVE OFF|ON

If set to **ON**, user names and host names specified with the options file **GROUP** and **HOST_GROUP** keywords, respectively, are treated as case insensitive.

By default, **GROUPCASEINSENSITIVE** is **OFF**, and user names and host names are treated as case sensitive.

HOST_GROUP

HOST_GROUP *group_name host_list*

Defines a group of hosts for use in **INCLUDE**, **INCLUDEALL**, **INCLUDE_ENTITLEMENT**, **EXCLUDE**, **EXCLUDEALL**, **EXCLUDE_ENTITLEMENT**, and **RESERVE** option lines. Multiple **HOST_GROUP** lines add all the specified hosts into the group.

Table 13-10: HOST_GROUP Terms

Term	Definition
group_name	Name of the group being defined. Host group names are case sensitive.
host_list	List of host names in that group. Names are case sensitive. Set the GROUPCASEINSENSITIVE options file keyword to turn on case insensitivity. See GROUPCASEINSENSITIVE .

To define the host group **Pacific** consisting of **tokyo**, **seattle**, and **auckland**:

```
HOST_GROUP Pacific tokyo seattle auckland
```

Anywhere a host name can be used in an options file, an IP address can be used instead.

INCLUDE

This option applies to concurrent licenses held in license files and trusted storage.

INCLUDE *feature[:keyword=value] type {name | group_name}*

Includes a user or predefined group of users in the list of who is allowed to use licenses for this feature. Any user who is not in an **INCLUDE** or **INCLUDEALL** statement is not allowed to use that feature. **EXCLUDE** supersedes **INCLUDE**; conflicts between the **EXCLUDE** list and the **INCLUDE** list are resolved by the **EXCLUDE** taking precedence.

Table 13-11: INCLUDE Terms

Term	Definition
feature	Name of the feature or package being affected.
keyword=value	Feature name modifier to denote a group of licenses. See Specifying Features for details.

Table 13-11: INCLUDE Terms

Term	Definition
type	One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See Specifying License Restrictions Using Type for details.
name	Name of an item of type <i>type</i> for which license usage is included.
group_name	Name of the group for which license usage is included.

To include user **bob** in the list of users able to use feature **f1**:

```
INCLUDE f1 USER bob
```

The include list is created from all the INCLUDEALL and INCLUDE lines in the options file.



Important: INCLUDE is required for USER_BASED or HOST_BASED features. The license administrator specifies which users are allowed to use the product, using INCLUDE in the options file, and the license limits the number of users that are INCLUDED. In a USER_BASED or HOST_BASED license model, users (or predefined groups of users) who are not listed with the INCLUDE keyword cannot check out a license.

INCLUDE_BORROW

This option is used for licenses held in license files. When licenses are available in trusted storage, normally activation is provided instead of BORROW.

```
INCLUDE_BORROW feature[:keyword=value] type {name | group_name}
```

Includes a user or predefined group of users in the list of who is allowed to borrow the BORROW feature. Anyone not in an INCLUDE_BORROW statement is not allowed to borrow licenses. EXCLUDE_BORROW supersedes INCLUDE_BORROW; conflicts between the EXCLUDE_BORROW list and the INCLUDE_BORROW list are resolved by the EXCLUDE_BORROW taking precedence.



Important: For USER_BASED or HOST_BASED features, a user or predefined group of users must be on both an INCLUDE list **and** an INCLUDE_BORROW list to borrow a feature.

Table 13-12: INCLUDE_BORROW Terms

Term	Definition
feature	Name of the feature being affected.
keyword=value	Feature name modifier to denote a group of licenses. See Specifying Features for details.
type	One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See Specifying License Restrictions Using Type for details.
name	Name of an item of type <i>type</i> for which license borrowing is included.
group_name	Name of the group for which license borrowing is included.

To include user **tom** in the list of users able to borrow feature **f1**:

```
INCLUDE_BORROW f1 USER tom
```

INCLUDE_ENTITLEMENT

This option only applies to licenses held in trusted storage.

```
INCLUDE_ENTITLEMENT entitlementId type {name | group_name}
```

Includes a user or predefined group of users in the list of who is allowed to activate the licenses contained in a fulfillment record held in trusted storage. **EXCLUDE_ENTITLEMENT** supersedes **INCLUDE_ENTITLEMENT**; conflicts between the **EXCLUDE_ENTITLEMENT** list and the **INCLUDE_ENTITLEMENT** list are resolved by the **EXCLUDE_ENTITLEMENT** taking precedence.

Table 13-13: INCLUDE_ENTITLEMENT Terms

Term	Definition
<i>entitlementId</i>	The entitlement Id originally used when requesting a license activation.
<i>type</i>	One of USER, HOST, GROUP, or HOST_GROUP. See Specifying License Restrictions Using Type for details.
<i>name</i>	Name of an item of type <i>type</i> for which license usage is included.
<i>group_name</i>	Name of the group to include.

To include the user **claire** in the list of users able to activate licenses provided in the fulfillment record specified by the entitlement Id AB456:

```
INCLUDE_ENTITLEMENT AB456 USER claire
```

INCLUDEALL

This option applies to concurrent licenses held in license files and trusted storage.

```
INCLUDEALL type {name | group_name}
```

Includes a user or predefined group of users in the list of who is allowed to use all features served by this vendor daemon.

Table 13-14: INCLUDEALL Terms

Term	Definition
type	One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See Specifying License Restrictions Using Type for details.
name	Name of an item of type <i>type</i> for which license usage is included.
group_name	Name of the group to include.

To allow the user **jane** to use all features served by this vendor daemon:

```
INCLUDEALL USER jane
```

The include list is created from all the INCLUDEALL and INCLUDE lines in the options file.

LINGER

This option applies to concurrent licenses held in license files and trusted storage.

```
LINGER feature[:keyword=value] seconds
```

A lingering license stays checked out for a specified period of time beyond its checkin or FLEXenabled application exit, whichever comes first. This option extends the default linger time configured by the software publisher in the FLEXenabled application.



Note: The software publisher must have enabled this feature in the FLEXenabled application for it to work. Contact your software publisher to find out if this feature is implemented.

Table 13-15: LINGER Terms

Term	Definition
feature	Name of the feature.
keyword=value	Feature name modifier to denote a group of licenses. See Specifying Features for details.
seconds	Number of seconds the license lingers. The software publisher sets a minimum value. If you specify a value for <i>seconds</i> that is smaller than the minimum, the minimum is used.

To set the linger value for feature `f1` to one hour (3600 seconds):

```
LINGER f1 3600
```

The actual linger time varies somewhat since the vendor daemon checks all lingering licenses just once per minute. If, however, a new license request is made that would otherwise be denied, a check of the lingering licenses is made immediately to attempt to satisfy the new request.

MAX

This option applies to concurrent licenses held in license files and trusted storage.

```
MAX num_lic feature[:keyword=value] type {name | group_name}
```

Limits usage for a group or user.

Table 13-16: MAX Terms

Term	Description
num_lic	Usage limit for this user or group.
feature	Feature or package this limit applies to.

Table 13-16: MAX Terms

Term	Description
keyword=value	Feature name modifier to denote a group of licenses. See Specifying Features for details.
type	One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See Specifying License Restrictions Using Type for details.
name	Name of an item of type <i>type</i> for which usage is limited.
group_name	Name of the group to limit.

For example, to limit the user **jan** to five licenses for feature **f1**, include the following line in the option file:

```
MAX 5 f1 USER jan
```

Queuing when Limit Defined by MAX is Reached

For vendor daemons version 10.8.x where x is greater than 7 and vendor daemons version 11.6.1 or greater, requests for licenses when the maximum has been reached will be queued if queuing is allowed by the application.

MAX_BORROW_HOURS

This option is used for licenses held in license files. When licenses are available in trusted storage, normally activation is provided instead of BORROW.

```
MAX_BORROW_HOURS feature[:keyword=value] num_hours
```

Changes the maximum period a license can be borrowed from that specified in the license file for *feature*. The new period must be less than that in the license file. If multiple MAX_BORROW_HOURS keywords appear in the options file, only the last one is applied to *feature*.

Table 13-17: MAX_BORROW_HOURS Terms

Term	Description
feature	Feature this borrow period applies to. The <i>feature</i> must have BORROW enabled.
keyword=value	Feature name modifier to denote a group of licenses. See Specifying Features for details.
num_hours	Number of hours in the new borrow period. This value must be less than that specified in the license file for feature (the default, if not specified, is 168 hours).

MAX_OVERDRAFT

This option applies to concurrent licenses held in license files and trusted storage.

```
MAX_OVERDRAFT feature[:keyword=value] num_lic
```

Limits OVERDRAFT license usage below the OVERDRAFT allowed by the license file.

Table 13-18: MAX_OVERDRAFT Terms

Term	Description
feature	Feature this limit applies to.
keyword=value	Feature name modifier to denote a group of licenses. See Specifying Features for details.
num_lic	Usage limit for this user or group.

NOLOG

NOLOG { IN | OUT | DENIED | QUEUED | UNSUPPORTED }

Suppresses logging the selected type of event in the debug log file.

Table 13-19: NOLOG Terms

Entry	Description
NOLOG IN	Turns off logging of checkins. Two separate NOLOG lines are required to turn off logging of checkouts and queued requests.
NOLOG DENIED NOLOG QUEUED	Turns off logging of checkouts and queued requests. License administrators use this option to reduce the size of the debug log file. However, it can reduce the usefulness of the debug log when debugging license server problems.
NOLOG UNSUPPORTED	Suppresses “UNSUPPORTED” messages in the debug log. This suppresses error messages in the debug log that report a failure due to the feature being unsupported.

See Also
[lmswitch](#)

REPORTLOG

REPORTLOG *[+]* *report_log_path*

REPORTLOG specifies the report log file for this vendor daemon. It is recommended preceding the *report_log_path* with a + character to append logging entries; otherwise, the file is overwritten each time the daemon is started.

On Windows, path names that include spaces have to be enclosed in double quotes. If *lmgrd* is started as a service, the default location for the report log file is the *c:\winnt\System32* folder unless a fully qualified path is specified.



Note: FLEXnet Manager is a separate product available from Acreesso Software, is used to process report log files. FLEXnet Manager processes only report log files, not debug log files.

Reporting on Projects with LM_PROJECT

The FLEXnet Manager report writer reports on projects. A project is set up by having all users working on the same project set their LM_PROJECT environment variable (or registry on Windows) to a string that describes the project. FLEXnet Manager groups usage by project, as defined by what LM_PROJECT was set to when the application was run.

See Also

[Configuring the License Server Manager as a Windows Service](#)
[Environment Variables](#)
[Report Log File](#)

RESERVE

This option applies to concurrent licenses held in license files and trusted storage.

`RESERVE num_lic feature[:keyword=value] type {name | group_name}`

Reserves licenses for a specific user.

Table 13-20: RESERVE Terms

Term	Description
num_lic	Number of licenses to reserve for this user or group.
feature	Feature or package this reservation applies to.
keyword=value	Feature name modifier to denote a group of licenses. See Specifying Features for details.
type	One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See Specifying License Restrictions Using Type for details.
name	Name of an item of type <i>type</i> for which license usage is reserved.
group_name	Name of group for which license usage is reserved.

To reserve one license of feature f1 for user mel:

```
RESERVE 1 f1 USER mel
```

If you want to reserve a license for *each* of several users or groups, you must use a separate RESERVE line for each user or group. If a package name is specified, all components that comprise the package are reserved.

The RESERVE keyword should not be used on packages and package suites that also use the SUITE_RESERVED option. The RESERVE keyword in the options file includes static information about the reservation policy. The SUITE_RESERVED option reserves a set of package components. Once one package component is checked out, all the other components are reserved for that same user. When the license keys are checked out by a user, the SUITE_RESERVED option within the Package will dictate additional reservation policy which can dynamically change depending on the product usage pattern. Because these options present conflicting reserve parameters, they can not be used together.



Note: Any licenses reserved for a user are dedicated to that user. Even when that user is not actively using the license, it is unavailable to other users. However, a RESERVED license does not cause usage to be reported by FLEXnet Manager if the license is not actually in use.

TIMEOUT

This option applies to concurrent licenses held in license files and trusted storage.

`TIMEOUT feature[:keyword=value] seconds`

Sets the time after which an inactive license is freed and reclaimed by the vendor daemon.



Note: The software publisher must have enabled this feature in the FLEXenabled application for it to work. Contact your software publisher to find out if this feature is implemented.

Table 13-21: TIMEOUT Terms

Term	Description
feature	Name of the feature.
keyword=value	Feature name modifier to denote a group of licenses. See Specifying Features for details.
seconds	Number of seconds after which inactive license is reclaimed. The software publisher sets a minimum value.If you specify a value for <i>seconds</i> that is smaller than the minimum, the minimum is used.

To set the timeout for feature f1 to one hour (3600 seconds):

`TIMEOUT f1 3600`

TIMEOUT checks in the licenses if the FLEXenabled application has been inactive for a period longer than the specified time period. The daemon declares a process inactive when it has not received heartbeats from it whereas an active FLEXenabled application sends heartbeats.

A TIMEOUT line must be present in the options file in order to take advantage of the this feature.

TIMEOUTALL

This option applies to concurrent licenses held in license files and trusted storage.

`TIMEOUTALL seconds`

Same as TIMEOUT, but applies to all features.

How the Vendor Daemon Uses the Options File

When the vendor daemon is started by `lmadmin` or `lmgrd`, the vendor daemon reads its options file. There is only one options file per vendor daemon and each vendor daemon needs its own options file. For any changes in an options file to take effect, the vendor daemon must read its options file. The `lmreread` utility causes the vendor daemon to reread its options file.

The `lmreread` utility enhanced in version 8.0 vendor daemon so that it causes the vendor daemon to reread the options file. If you are using earlier versions, the vendor daemon must be stopped and restarted in order for the options file to be reread.

Rules of Precedence in Options Files

Rules of precedence take effect when `INCLUDE` and `EXCLUDE` statements are combined in the same options file and control access to the same feature (in license files) or fulfillment record (in trusted storage). The following define the precedence when both types of statements appear together:

- If there is only an `EXCLUDE` list, everyone who is not on the list is allowed to use the feature.
- If there is only an `INCLUDE` list, only those users on the list are allowed to use the feature.
- If neither list exists, everyone is allowed to use the feature.
- The `EXCLUDE` list is checked before the `INCLUDE` list; someone who is on both lists is not allowed to use the feature.

Once you create an `INCLUDE` or `EXCLUDE` list, everyone else is *implicitly* outside the group. This feature allows you, as a license administrator, the ability to control licenses without having to *explicitly* list each user that you wish to allow or deny access to. In other words, there are two approaches; you either:

- Give most users access and list only the exceptions, or
- Severely limit access and list only the those users that have access privileges

Options File Examples

The following information gives some examples of options files intended to illustrate ways to effectively control access to your licenses.

Simple Options File Example

```
RESERVE 1 compile USER robert
RESERVE 3 compile HOST mainline
EXCLUDE compile USER lori
NOLOG QUEUED
```

This options file restricts the use of concurrent licenses as follows:

- Reserves one license for the feature **compile** for the user **robert**.

- Reserves three licenses for the feature **compile** for anyone on the system with the host name **mainline**.
- Prevents the user **lori** from using the **compile** feature on any system on the network.
- Causes QUEUED messages to be omitted from the debug log file.

The sum total of the licenses reserved must be less than or equal to the number of licenses specified in the FEATURE line. In the example above, there must be a minimum of four licenses on the **compile** FEATURE line. If fewer licenses are available, only the first set of reservations (up to the license limit) is used.

If this data were in file `/a/b/sampled/licenses/sampled.opt`, then modify the license file VENDOR line as follows:

```
VENDOR sampled /etc/sampled /sample_app/sampled/licenses/sampled.opt
```

Limiting Access for Multiple Users

Each INCLUDE, INCLUDEALL, INCLUDE_BORROW, INCLUDE_ENTITLEMENT, EXCLUDE, EXCLUDEALL, EXCLUDE_BORROW, EXCLUDE_ENTITLEMENT, MAX, and RESERVE line must have a single user name (or group) listed. To affect more than one user name create a GROUP. For example to exclude **bob**, **howard**, and **james** from using the feature called **toothbrush**, create the following options file:

```
EXCLUDE toothbrush USER bob
EXCLUDE toothbrush USER howard
EXCLUDE toothbrush USER james
```

However, there is an easier way. Create a GROUP and exclude the list of users from using the feature. Like the previous example, the following options file excludes **bob**, **howard**, and **james** from using the feature called **toothbrush**:

```
# First define the group "Hackers"
GROUP Hackers bob howard james
# Then exclude the group
EXCLUDE toothbrush GROUP Hackers
```

Now when you want to allow or deny access to any feature to that group, you have an alias list to make it simple.

Use HOST_GROUP to allow, deny, or reserve licenses for multiple hosts. For example, to exclude all users logged in on the hosts **fred** and **barney** from using a feature called **f1**, add these lines to your options file:

```
HOST_GROUP writers fred barney
EXCLUDE f1 HOST_GROUP writers
```

See Also

[HOST_GROUP](#) for more information about defining groups

EXCLUDE Example

```
#First Define the group "painters"
GROUP painters picasso mondrian klee
EXCLUDE spell GROUP painters
```

```
EXCLUDE spell USER bob
EXCLUDE spell INTERNET 123.123.123.*
```

This options file:

- Prevents the users **picasso**, **mondrian**, and **klee** from using the feature **spell** on any system on the network.
- Prevents the user **bob** from using the feature **spell** on any system on the network.
- Prevents any user logged into a host with an IP address in the range 123.123.123.0 through 123.123.123.255 from using the feature **spell**.
- Allows any other user, as long as they are not on the excluded IP addresses, *and* they are not a member of the **painters** GROUP, *and* they are not **bob**, to use feature **spell** (by implication).

Note that **bob** could have been added to the group **painters**. However, **painters** might be used for some other purpose in the future so the license administrator chose to handle **bob** as a special case here. In this case, the two EXCLUDE statements concatenate to create a list of four users.

EXCLUDE_ENTITLEMENT Example

```
#First Define the group "admin"
GROUP admin johns adrianp maryt
EXCLUDE_ENTITLEMENT qf573k GROUP admin
EXCLUDE_ENTITLEMENT qf573k USER bob
EXCLUDE_ENTITLEMENT qf573k HOST cordelia
```

This options file:

- Prevents the users johns, adrianp, and maryt from activating any licenses contained in the fulfillment record obtained using the entitlement Id qf573k on any system on the network.
- Prevents the user bob from activating any licenses contained in the fulfillment record obtained using the entitlement Id qf573k on any system on the network.
- Prevents any user on the system called **cordelia** from activating any licenses contained in the fulfillment record obtained using the entitlement Id qf573k.
- By implication allows any other users on any system other than **cordelia** to activate the licenses contained in the fulfillment record obtained using the entitlement Id qf573k.

INCLUDE Example

```
INCLUDE paint USER picasso
INCLUDE paint USER mondrian
INCLUDE paint HOST bigbrush
```

This options file:

- Allows the user **picasso** to use the feature **paint** on any system on the network.
- Allows the user **mondrian** to use the feature **paint** on any system on the network.
- Allows any user, as long as they are on the host **bigbrush**, to use feature **paint**.

- Denies access to the feature **paint** to anyone except **picasso**, **mondrian**, or anyone from the host **bigbrush** (by implication).

INCLUDE_ENTITLEMENT Example

```
INCLUDE_ENTITLEMENT gy7210 USER tom  
INCLUDE_ENTITLEMENT gy7210 USER anthony  
INCLUDE_ENTITLEMENT gy7210 HOST jupiter
```

This options file:

- Allows the user tom to activate any licenses contained in the fulfillment record obtained using the entitlement Id gy7210 on any system on the network.
- Allows the user anthony to activate any licenses contained in the fulfillment record obtained using the entitlement Id gy7210 on any system on the network.
- Allows any user, as long as they are on the host jupiter to activate any licenses contained in the fulfillment record obtained using the entitlement Id gy7210.
- By implication denies the activation of any licenses contained in the fulfillment record obtained using the entitlement Id gy7210 to anyone except tom, anthony, or someone on the host jupiter.