

# 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
- Control the automatic rereading of licenses

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 4000 characters. The \ character is the line-continuation character.



---

**Note** • The following lists changes in the options file for FlexNet Publisher versions:

- PROJECT identification (set by LM\_PROJECT) in options file was introduced in the version 7.0 vendor daemon.

- Option file control for licenses held in fulfillment records in trusted storage was introduced in the 11.3 vendor daemon.
- AUTOMATIC\_REREAD keyword introduced in the version 11.7 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**, was introduced in the version 6 vendor daemon.

## Options File Syntax

The following is an overview of the available options. See [Options File Examples](#) for examples and additional information. Each line of the file controls one option.

**Table 13-1** • Option Keywords

| Option Keyword                | Description                                                                                                |
|-------------------------------|------------------------------------------------------------------------------------------------------------|
| <b>AUTOMATIC_REREAD</b>       | Turn off automatic reread of licenses at midnight.                                                         |
| <b>ACTIVATION_LOWWATER</b>    | Controls the number of licenses that cannot be borrowed or transferred.                                    |
| <b>ACTIVATION_EXPIRY_DAYS</b> | Controls the activation request based on the expiration dates mentioned in options file during activation. |
| <b>BORROW_LOWWATER</b>        | Set the number of BORROW licenses that cannot be borrowed.                                                 |

Table 13-1 • Option Keywords

| Option Keyword               | Description                                                                                                     |
|------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <b>DAEMON_SELECT_TIMEOUT</b> | Defines the vendor daemon's timeout limit.                                                                      |
| <b>DEBUGLOG</b>              | Writes debug log information for this vendor daemon to the specified file (version 8.0 or later vendor daemon). |
| <b>EXCLUDE</b>               | Deny a user access to a feature.                                                                                |
| <b>EXCLUDE_BORROW</b>        | Deny a user the ability to borrow BORROW licenses.                                                              |
| <b>EXCLUDE_ENTITLEMENT</b>   | Deny a user the ability to activate licenses held in a fulfillment record in trusted storage.                   |
| <b>EXCLUDEALL</b>            | Deny a user access to <i>all</i> features served by this vendor daemon.                                         |
| <b>FQDN_MATCHING</b>         | Sets the level of host name matching.                                                                           |
| <b>GROUP</b>                 | Define a group of users for use with any options.                                                               |
| <b>GROUPCASEINSENSITIVE</b>  | Sets case sensitivity for user and host lists specified in GROUP and HOST_GROUP keywords.                       |
| <b>HOST_GROUP</b>            | Define a group of hosts for use with any options (version 4.0 or later).                                        |
| <b>INCLUDE</b>               | Allow a user to use a feature.                                                                                  |
| <b>INCLUDE_BORROW</b>        | Allow a user to borrow BORROW licenses.                                                                         |
| <b>INCLUDE_ENTITLEMENT</b>   | Allow a user to activate licenses held in a fulfillment record in trusted storage.                              |
| <b>INCLUDEALL</b>            | Allow a user to use <i>all</i> features served by this vendor daemon.                                           |
| <b>LINGER</b>                | Allow a user to extend the linger time for a feature beyond its check in.                                       |
| <b>MAX</b>                   | Limit usage for a particular feature/group—prioritizes usage among users.                                       |
| <b>MAX_BORROW_HOURS</b>      | Changes the maximum borrow period for the specified feature.                                                    |
| <b>MAX_OVERDRAFT</b>         | Limit overdraft usage to less than the amount specified in the license.                                         |
| <b>NOLOG</b>                 | Turn off logging of certain items in the debug log file.                                                        |
| <b>REPORTLOG</b>             | Specify that a report log file suitable for use by the FlexNet Manager license usage reporting tool be written. |
| <b>RESERVE</b>               | Reserve licenses for a user or group of users/hosts.                                                            |

**Table 13-1 • Option Keywords**

| Option Keyword    | Description                                                                                |
|-------------------|--------------------------------------------------------------------------------------------|
| <b>TIMEOUT</b>    | Specify idle timeout for a feature, returning it to the free pool for use by another user. |
| <b>TIMEOUTALL</b> | Set timeout on all features.                                                               |

## Comments

Include comments in your options file by starting each comment line with the hash symbol, #.

## 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.

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.



**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"

## 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.

For the MAX option, you can also specify ALL\_USERS as a value for the USER type. See [MAX](#) for details.

- **GROUP**—Name of the group of users executing the FlexEnabled application. (Before you can restrict an option by a specific group, the options file must also include a [GROUP](#) option that defines the group.)

For the MAX option, you can also specify ALL\_GROUPS as a value for the GROUP type. See [MAX](#) for details.

- **HOST**—System host name or IP address where the application is executing. Host names are not case sensitive. 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.
- **PROJECT**—LM\_PROJECT environment variable set by the user who is executing the FlexEnabled application. Project names are case sensitive. See [Chapter 24, "Environment Variables,"](#) for more information about this variable.

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.

## AUTOMATIC\_REREAD

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

`AUTOMATIC_REREAD OFF|ON`

Controls the automatic rereading of license files and trusted storage when any features are found to have expired. The default when this option is not set is that at midnight each day a check of each license is made to determine if it has expired. When any license is found to have expired, all license files and trusted storage are reread.

To turn off this automatic reread at midnight, enter `AUTOMATIC_REREAD OFF` in the options file.

## ACTIVATION\_LOWWATER

This option allows license administrator to control the number of hybrid and activatable license that cannot be borrowed or transferred. It works with a hybrid and activatable license in server side trusted storage.

Syntax

`ACTIVATION_LOWWATER entitlementID count`  
`ACTIVATION_LOWWATER entitlementID:FID=fulfillmentID count`

For example, the server has an entitlement AAAA with 10 hybrid counts.

The following line in the options file allow only 4 licenses from entitlement AAAA to be borrowed or transferred to another license server.

```
ACTIVATION_LOWWATER AAAA 6
```

**Table 13-2 • ACTIVATION\_LOWWATER Terms**

| Term                               | Description                                                                                           |
|------------------------------------|-------------------------------------------------------------------------------------------------------|
| <b>entitlementID</b>               | The entitlement Id originally used when requesting license activation.                                |
| <b>Hybrid or Activatable count</b> | Number of licenses that cannot be borrowed/transferred.                                               |
| <b>fulfillmentID (Optional)</b>    | A unique identity for this fulfillment record. Used to specify the fulfillment record to be returned. |



**Note** • If more than one `ACTIVATION_LOWWATER` option for the same `EntitlementID` and `fulfillmentID` is specified in the options file, only the first one will be considered and the rest of duplicate entries will be discarded.

## ACTIVATION\_EXPIRY\_DAYS

This option enables license administrators to control the activation request based on the expiration dates mentioned in the options file during activation. The publisher controls the activations based on the number of expiration days. If the expiration date is greater than the date set in the options file, the functionality is not successful.

`ACTIVATION_EXPIRY_DAYS entitlement ID days`

ACTIVATION\_EXPIRY\_DAYS entitlementID:FID=fulfillmentID days

**Table 13-3 • ACTIVATION\_EXPIRY\_DAYS**

| Term                                | Description                                                                                           |
|-------------------------------------|-------------------------------------------------------------------------------------------------------|
| <b>entitlementID</b>                | The entitlement Id originally used when requesting license activation.                                |
| <b>fulfillmentID<br/>(Optional)</b> | A unique identity for this fulfillment record. Used to specify the fulfillment record to be returned. |

For example, if there is more than one entry in the options file, the functionality works fine with respect to the first line given in the options file.

ACTIVATION\_EXPIRY\_DAYS ENTL-EZCALC 3

As per the example, if the current date is 11-December-2016 then the client can activate the license untill 13-dec-2016. In case of normal activation without the options file, the client can borrow the license beyond 13-dec-2016.



**Note** • If more than one *ACTIVATION\_EXPIRY\_DAYS* option for the same *EntitlementID* and *FulfillmentID* with different count of days is mentioned in the options file, then the functionality always considers the first line and ignores subsequent lines.

## 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 13-4 • BORROW\_LOWWATER Terms**

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

For example, 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 for this feature to be borrowed:

BORROW\_LOWWATER f1 3

## DAEMON\_SELECT\_TIMEOUT

DAEMON\_SELECT\_TIMEOUT *value\_in\_seconds*

Specifies the threshold for the amount of time (in seconds) that can elapse before a vendor-daemon connection experiences a timeout. The maximum value is **60**; the minimum is **1**.



**Important** • You can define the DAEMON\_SELECT\_TIMEOUT option only when the publisher has enabled it (that is, set the vendor variable `ls_support_custom_daemon_select_timeout` to **1**). Best practice is for publishers to enable DAEMON\_SELECT\_TIMEOUT only for environments that have limited bandwidth. If this option is not enabled, the timeout value for the vendor-daemon connection is 1 second, by default.

The following rules apply to DAEMON\_SELECT\_TIMEOUT:

- A value over 60 is always considered **60**.
- A value under 1 is always considered **1**.
- The default value is **1**.

## 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 the license server manager 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 debug log file is by default generated in the path specified in the options file. It is best practice to set the location of `debug_log_path` to a ProgramData sub-folder, since this location has user-write permission by default, which is needed when the license server runs as a service with LocalService permission.

### 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 13-5 • 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 <a href="#">“Specifying Features”</a> for details.                                        |
| <i>type</i>          | One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See <a href="#">“Specifying License Restrictions Using Type”</a> 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. Group names are case sensitive.                                                                                      |

For example, the following option syntax excludes 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 13-6 • 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 <a href="#">Specifying Features</a> for details.                                        |
| <i>type</i>          | One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See <a href="#">Specifying License Restrictions Using Type</a> 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. Group names are case sensitive.                                                                     |

For example, the following option syntax excludes 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 13-7 • 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 <a href="#">Specifying License Restrictions Using Type</a> 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. Group names are case sensitive.                                                        |

For example, the following option syntax excludes 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 13-8 • EXCLUDEALL Terms**

| Term                     | Description                                                                                                                                      |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b><i>type</i></b>       | One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See <a href="#">Specifying License Restrictions Using Type</a> for details. |
| <b><i>name</i></b>       | Name of an item of type <i>type</i> for which license usage is excluded.                                                                         |
| <b><i>group_name</i></b> | Name of the group to exclude. Group names are case sensitive.                                                                                    |

For example, the following option syntax excludes 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 13-9 • FQDN\_MATCHING Terms**

| Term                  | Description                                                                                                                                                   |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b><i>exact</i></b>   | 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.                       |
| <b><i>lenient</i></b> | The host name sent by the application needs to 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 13-10 shows the outcome of matching attempts between HOST type-specifiers in the options file and host names sent by the application.

**Table 13-10 • Host Name Matching Matrix**

| Options File Settings        |                                            | Application configured for FQDN—sends myhost.abc.com | Application not configured for FQDN—sends myhost |
|------------------------------|--------------------------------------------|------------------------------------------------------|--------------------------------------------------|
| <b>FQDN_MATCHING exact</b>   | INCLUDE <i>feature</i> HOST myhost         | no                                                   | yes                                              |
|                              | INCLUDE <i>feature</i> HOST myhost.abc.com | yes                                                  | no                                               |
| <b>FQDN_MATCHING lenient</b> | INCLUDE <i>feature</i> HOST myhost         | yes                                                  | yes                                              |
|                              | INCLUDE <i>feature</i> 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 13-11 • GROUP Terms**

| Term                     | Description                                                                                                                                                                                                       |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b><i>group_name</i></b> | Name of the group being defined. Group names are case sensitive.                                                                                                                                                  |
| <b><i>user_list</i></b>  | 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 <a href="#">GROUPCASEINSENSITIVE</a> . |

To create a large user group, define several GROUP lines each containing up to the maximum of 4,000 characters. All the users will be placed in a single group: Multiple GROUP lines for the same group name add all the specified users into the group.

For example, the following option syntax defines 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 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 13-12 • HOST\_GROUP Terms**

| Term              | Definition                                                                                                                                                                                           |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>group_name</b> | Name of the group being defined. Host group names are case sensitive.                                                                                                                                |
| <b>host_list</b>  | List of host names in that group. Names are case sensitive. Set the <code>GROUPCASEINSENSITIVE</code> options file keyword to turn on case insensitivity. See <a href="#">GROUPCASEINSENSITIVE</a> . |

For example, the following option syntax defines 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 13-13 • INCLUDE Terms**

| Term                 | Definition                                                                                                                                                                                                                                         |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>feature</b>       | Name of the feature or package being affected.                                                                                                                                                                                                     |
| <b>keyword=value</b> | Feature name modifier to denote a group of licenses. See <a href="#">Specifying Features</a> for details.                                                                                                                                          |
| <b>type</b>          | One of <code>USER</code> , <code>HOST</code> , <code>DISPLAY</code> , <code>INTERNET</code> , <code>PROJECT</code> , <code>GROUP</code> , or <code>HOST_GROUP</code> . See <a href="#">Specifying License Restrictions Using Type</a> for details. |
| <b>name</b>          | Name of an item of type <code>type</code> for which license usage is included.                                                                                                                                                                     |
| <b>group_name</b>    | Name of the group for which license usage is included. Group names are case sensitive.                                                                                                                                                             |

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.



**Note** • INCLUDE is required for USER\_BASED or HOST\_BASED features. The license administrator specifies which users are allowed to use the product, via INCLUDE, 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.

**Table 13-14** • INCLUDE\_BORROW Terms

| Term                 | Definition                                                                                                                                       |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>feature</b>       | Name of the feature and package being affected. Specify the package name and the feature name or specify only the package name.                  |
| <b>keyword=value</b> | Feature name modifier to denote a group of licenses. See <a href="#">Specifying Features</a> for details.                                        |
| <b>type</b>          | One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See <a href="#">Specifying License Restrictions Using Type</a> for details. |
| <b>name</b>          | Name of an item of type <i>type</i> for which license borrowing is included.                                                                     |
| <b>group_name</b>    | Name of the group for which license borrowing is included. Group names are case sensitive.                                                       |

For example, the following option syntax includes user tom in the list of users able to borrow feature f1:

```
INCLUDE_BORROW f1 USER tom
```



**Note** • 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.

# 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 13-15 • INCLUDE\_ENTITLEMENT Terms**

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

For example, the following option syntax includes 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
```

## Example for Include variations with hybrid trusted storage:

This example demonstrates the behavior of **INCLUDE**, **INCLUDE\_BORROW** and **INCLUDE\_ENTITLEMENT** on *feature F1*. *F1* originates from a served trusted storage hybrid license with *EntitlementID ENTLID1*.

**Table 13-16 • Example for INCLUDE, INCLUDE\_BORROW, INCLUDE\_ENTITLEMENT**

| Option file         | Contains          | Behavior                                                                                                                     |
|---------------------|-------------------|------------------------------------------------------------------------------------------------------------------------------|
| INCLUDE             | F1 USER John      | concurrent checkout limited to John? Yes<br>certificate borrow limited to John? Yes<br>activation borrow limited to John? No |
| INCLUDE_BORROW      | F1 USER John      | concurrent checkout limited to John? No<br>certificate borrow limited to John? Yes<br>activation borrow limited to John? No  |
| INCLUDE             | F1 USER John      | concurrent checkout limited to John? Yes                                                                                     |
| INCLUDE_ENTITLEMENT | ENTLID1 USER John | certificate borrow limited to John? Yes<br>activation borrow limited to John? Yes                                            |

**Table 13-16** • Example for INCLUDE, INCLUDE\_BORROW, INCLUDE\_ENTITLEMENT

| Option file         | Contains          | Behavior                                                                                                                    |
|---------------------|-------------------|-----------------------------------------------------------------------------------------------------------------------------|
| INCLUDE_ENTITLEMENT | ENTLID1 USER John | concurrent checkout limited to John? No<br>certificate borrow limited to John? No<br>activation borrow limited to John? Yes |

Another example: *ENTLID12* contains hybrid features *F1* and *F2*. The following options file allows only John to perform concurrent checkout, certificate borrow or activation borrow on all *ENTLID12*'s features:

```
INCLUDE F1 USER John
```

```
INCLUDE F2 USER John
```

```
INCLUDE_ENTITLEMENT ENTLID12 USER John
```



**Note** • activation borrow always activates all features from an entitlement

## 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 13-17** • INCLUDEALL Terms

| Term              | Definition                                                                                                                                       |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>type</b>       | One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See <a href="#">Specifying License Restrictions Using Type</a> for details. |
| <b>name</b>       | Name of an item of type <i>type</i> for which license usage is included.                                                                         |
| <b>group_name</b> | Name of the group to include. Group names are case sensitive.                                                                                    |

For example, the following option syntax allows 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. The linger time may have been configured by the software publisher in the FlexEnabled application. When this is the case, then the longer linger time is applied. Thus you can set a longer linger time than configured by the software publisher but not shorten the linger time.

**Table 13-18 • LINGER Terms**

| Term                        | Definition                                                                                                                                                                           |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b><i>feature</i></b>       | Name of the feature.                                                                                                                                                                 |
| <b><i>keyword=value</i></b> | Feature name modifier to denote a group of licenses. See <a href="#">Specifying Features</a> for details.                                                                            |
| <b><i>seconds</i></b>       | 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. |

For example, the following option syntax sets 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. Also if 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 13-19 • MAX Terms**

| Term                        | Description                                                                                                                                      |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b><i>num_lic</i></b>       | Usage limit for this user or group.                                                                                                              |
| <b><i>feature</i></b>       | Feature or package this limit applies to.                                                                                                        |
| <b><i>keyword=value</i></b> | Feature name modifier to denote a group of licenses. See <a href="#">Specifying Features</a> for details.                                        |
| <b><i>type</i></b>          | One of USER, HOST, DISPLAY, INTERNET, PROJECT, GROUP, or HOST_GROUP. See <a href="#">Specifying License Restrictions Using Type</a> for details. |

Table 13-19 • MAX Terms

| Term                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b><i>name</i></b>       | <p>Name of an item of type <i>type</i> for which usage is limited.</p> <p>To avoid having to repeat the MAX line multiple times to restrict all users, you can use one MAX line that specifies ALL_USERS as the value for the type USER (for example, USER ALL_USERS). See for <a href="#">USER and GROUP Examples</a> more information.</p>                                                                                                    |
| <b><i>group_name</i></b> | <p>Name of the group to limit. Group names are case sensitive. (The options file must include a GROUP line that defines this group.)</p> <p>To avoid having to repeat the MAX line multiple times to restrict all groups defined in the options file, you can use one MAX line that specifies ALL_GROUPS as the value for the type GROUP (for example, GROUP ALL_GROUPS). See for <a href="#">USER and GROUP Examples</a> more information.</p> |

## USER and GROUP Examples

The following are examples of using the USER and GROUP types for the MAX option.



**Note** • Never position a MAX line that restricts USER *user\_name* for a specific feature after a MAX line that restricts USER ALL\_USERS for the same feature. Likewise, never position a MAX line that restricts GROUP *group\_name* for a specific feature after a MAX line that restricts GROUP ALL\_GROUPS for the same feature. The positioning invalidates the restriction in the later line.

- To limit the user jan to five licenses for feature f1, include the following line in the options file:  
MAX 5 f1 USER jan
- To limit all users to to five licenses for feature f1, include the following line:  
MAX 5 f1 USER ALL\_USERS
- To limit the user jan to five licenses for feature f1 and all other users to four licenses for the same feature, include the following lines:  
MAX 5 f1 USER jan  
MAX 4 f1 USER ALL\_USERS
- To limit group DEV to three license for feature f2 and all other groups to one license for the same feature, include these lines:  
MAX 3 f2 GROUP DEV  
MAX 1 f2 GROUP ALL\_GROUPS
- If you attempt to position a MAX line that specifies ALL\_USERS (or ALL\_GROUPS) before a MAX line that specifies a single user (or single group), FlexNet Publisher cannot enforce the restriction in the second line because the user or group in the second line (in this case, the DEV group) is a part of ALL\_USERS or ALL\_GROUPS in the first line.  
For example, the following line positioning is *invalid*:  
MAX 1 f2 GROUP ALL\_GROUPS  
MAX 3 f2 GROUP DEV

## Queueing Behavior When Requested Licenses Exceed MAX Limit

For current version vendor daemons, if queuing is allowed by the application, requests for licenses that exceed the limit set by the MAX keyword will be denied. From the above example, if the user jan requests six licenses for feature f1, the request will be denied. Requests from users or groups within the MAX limit that exceed the number of available licenses will be queued. For example, if the license file includes ten licenses for feature f1 and nine of those licenses are already checked out, a request from the user jan for two licenses will be queued.

## 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 13-20 • MAX\_BORROW\_HOURS Terms

| Term                 | Description                                                                                                                                                          |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>feature</i>       | Feature this borrow period applies to. The <i>feature</i> must have BORROW enabled.                                                                                  |
| <i>keyword=value</i> | Feature name modifier to denote a group of licenses. See <a href="#">Specifying Features</a> for details.                                                            |
| <i>num_hours</i>     | 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 13-21 • MAX\_OVERDRAFT Terms

| Term                 | Description                                                                                               |
|----------------------|-----------------------------------------------------------------------------------------------------------|
| <i>feature</i>       | Feature this limit applies to.                                                                            |
| <i>keyword=value</i> | Feature name modifier to denote a group of licenses. See <a href="#">Specifying Features</a> for details. |
| <i>num_lic</i>       | 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 13-22 • NOLOG Terms**

| Entry                                      | Description                                                                                                                                                                                                                         |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NOLOG IN</b>                            | Turns off logging of checkins. Two separate NOLOG lines are required to turn off logging of checkouts and queued requests.                                                                                                          |
| <b>NOLOG OUT</b>                           | Turns off logging of checkouts.                                                                                                                                                                                                     |
| <b>NOLOG DENIED</b><br><b>NOLOG QUEUED</b> | 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. |
| <b>NOLOG UNSUPPORTED</b>                   | 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 current working directory unless a fully qualified path is specified.



**Note •** FlexNet Manager is a separate product available from Flexera Software, it is used to process report log files. FlexNet Manager processes only report log files, not debug log files. On Windows, It is best practice to set the location of report\_log\_path to a ProgramData sub-folder, since this location has user-write permission by default, which is needed when the license server runs as a service with LocalService permission

## 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 13-23 • RESERVE Terms**

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

For example, the following option syntax reserves 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. The total number of reservations supported per options file is 10000.



**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 13-24 • TIMEOUT Terms**

| Term                 | Description                                                                                                                                                                                                                                  |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>feature</b>       | Name of the feature.                                                                                                                                                                                                                         |
| <b>keyword=value</b> | Feature name modifier to denote a group of licenses. See <a href="#">Specifying Features</a> for details.                                                                                                                                    |
| <b>seconds</b>       | 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 or greater than the minimum, the greater value is used. |

For example, the following option syntax sets 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 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 **Imrread** 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 does the following:

- 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.

