MARC/Link-Pro Version 14+

This version works with:

Pro/ENGINEER version 14.0 and above. It has been tested with Pro/ENGINEER versions 16.0 and 17.0. It does not work with Pro/ENGINEER versions 13.0 and below.

After the correct menu files are copied into the current Mentat directory, this version should work with Mentat release 2.3 and above.

The following lines have to be added to user's local `.cshrc` or Pro/ENGINEER command file:

```
setenv DIR "your_mentat_installation_location"
setenv MANCHOR $DIR/security/marcmgc.'hostname'
setenv LD_LIBRARY_PATH $DIR/bin
```

Alternatively the following lines should be added to user's local `.profile`:

```
DIR="your_mentat_installation_location"
MANCHOR=$DIR/security/marcmgc.'hostname'
export MANCHOR
LD_LIBRARY_PATH=$DIR/bin
export LD_LIBRARY_PATH
```

Printed in the USA.
March, 1997

Contents

Welcome to FLEX*lm*

FLEX*lm*® is the de facto standard network license manager used by over 1000 leading software vendors to control the use of their software products. If you are a system administrator or user, chances are one or more of the products currently on your network is licensed by FLEX*lm*.

This manual explains FLEX*lm* from an end-user standpoint and how to use the command-line tools which are part of the standard FLEX*lm* distribution kit. GLOBE*trotter* Software also provides the FLEX*admin*™ asset management tool for more advanced license server control and reporting. Please contact GLOBE*trotter* Software (info@globes.com) for more information about FLEX*admin*.

Keep in mind that certain topics (such as password encryption) are vendor-specific and proprietary so we cannot document them in any detail. Also, because FLEX*lm* does not enforce a particular licensing strategy, each vendor's implementation can have subtle differences. If you do not find out what you need to know here, you should contact your vendor's technical support group.

Versions of FLEX*lm*

This manual covers features of interest to license administrators and end-users in FLEX*lm* versions 1.0 through 5.11.

How to Use This Manual

This manual is written for two different audiences: the license administrator and the end user.

License Administrator

If you are a license administrator, read these chapters:

This chapter:	Explains:
Introduction and Overview	FLEX*lm* basics: license and vendor daemons; the license file; configuring FLEX*lm*; the license request process.
The License File	The license file format; setting the path at start-up; different types of licensing policies.
Multiple License Files	Using license files from independent software vendors.

This chapter:	Explains:
Selecting Server Nodes	Selecting which machines will run the license servers, resources required by the servers, multiple servers, quorums, and redundant servers.
The Options File	Creating and editing the options file.
License Administration Tools	Managing FLEX*lm* using GLOBE*trotter*-supplied utilities.

In addition, you can refer to Appendix B, "Troubleshooting Guide" on page 53, which contains a list of common problems and their solutions and Appendix C, "Frequently Asked Questions" on page 59.

End User

If you are an end user, read these chapters:

This chapter:	Explains:
Introduction and Overview	FLEX*lm* basics: license and vendor daemons; the license file; configuring FLEX*lm*; the license request process.
The License File	The license file format; setting the path at start-up; different types of licensing policies.

In addition, you can refer to Appendix B, "Troubleshooting Guide" on page 53, which contains a list of common problems and their solutions.

Related Documents from GLOBE*trotter* Software

FLEXadmin™*/FLEXwrap*™ *Users Guide* - describes both the FLEX*admin* license administration tools for managing FLEXlm-enabled applications, and *FLEXwrap* for administrators who wish to provide license management for applications shipped without embedded license management.

FLEXlm Programmers Guide and *FLEXlm Reference Manual* - for programmers responsible for incorporating FLEX*lm* into their products.

Chapter 1

Introduction and Overview

This chapter explains the basics of floating (network) licensing, and gives a quick overview of the components of FLEX*lm*. It explains where license administrators have control and where end-users have control. A section called "Getting Started Checklist" on page 4 tells both license administrators and end-users how to start managing FLEX*lm*.

1.1 Introduction to FLEX*lm*

FLEX*lm* is the most popular license manager used in the software industry. FLEX*lm* is best known for its ability to allow software licenses to be available (or float) anywhere on a network, instead of being tied to specific machines. Floating licensing benefits both users and license administrators. Users can make more efficient use of fewer licenses by sharing them on the network. License administrators can control who uses the licensed application, and the node(s) where the licenses will be available. See Section 2.4, "Types of License Files," on page 17 for details about the different licensing models supported by FLEX*lm*.

1.1.1 FLEX*lm* Components

The four main components of FLEX*lm* are:

- license manager daemon
- vendor daemon
- license file
- application program

THE LICENSE MANAGER DAEMON (*LMGRD*)

The *license manager daemon* (*lmgrd*) handles the initial contact with the client application programs, passing the connection on to the appropriate vendor daemon. It also starts and restarts the vendor daemons. FLEX*lm* permits multiple redundant license manager daemons on three server nodes, allowing you to make your license available if any two out of the three server nodes is running. Redundancy can be achieved with 3-server redundant servers, or by using a license file list with any number of servers.

Note—*lmgrd* is not present on VMS or Netware systems.

THE VENDOR DAEMON

In FLEX*lm*, licenses are granted by running processes (unless they're node locked, uncounted, in which case they need only read the license file to run). There is one process for each vendor who has a FLEX*lm*-licensed product on the network. This process is called the *vendor daemon*. The vendor daemon keeps track of how many licenses are checked out, and who has them. If the vendor daemon terminates for any reason, all users lose their licenses (though this does not mean the applications suddenly stop running). Users normally regain their license automatically when *lmgrd* restarts the vendor daemon, though they may exit if the vendor daemon remains unavailable.

Client programs communicate with the vendor daemon, usually through TCP/IP network communications. The client application and the daemon processes (the license server) can run on separate nodes on your network, across any size wide-area network. Also, the format of the traffic between the client and the vendor daemon is machine-independent, allowing for heterogenous networks. This means the license server and the computer running an application can be either different hardware platforms or even different operating systems (Windows and Unix, for example).

THE LICENSE FILE

Licensing data is stored in a text file called the license file. The license file is created by the software vendor, and edited and installed by the license administrator. It contains information about the server nodes and vendor daemons, and at least one line of data (called FEATURE or INCREMENT lines) for each licensed product. Each FEATURE line contains a license key based on the data in that line, the hostids specified in the SERVER lines, and other vendor-specific data.

In some environments, the licensing information for several vendors may be combined into a single license file. The default location is:

```
/usr/local/flexlm/licenses/license.dat (Unix)
C:\flexlm\license.dat (Windows, Windows/NT, OS/2)
SYS$COMMON:[SYSMGR]flexlm.dat (VMS)
SYS:\SYSTEM\flexlm\license.dat (Netware)
```

End-users can usually override this location by setting the environment variable `LM_LICENSE_FILE` to point elsewhere, or by following instructions supplied with the licensed application. If your site has software from multiple vendors with incompatible license files (due to different sets of servers), you can keep the data in separate files and set the `LM_LICENSE_FILE` variable to reference multiple files.

It's strongly recommended that you keep a copy or link (on Unix) of the license file in the vendor's "default" location, so that users will not need to set LM_LICENSE_FILE to run their applications. For details, see Chapter 2, "The License File" on page 7.

THE APPLICATION PROGRAM

The application program using FLEX*lm* is linked with the program module (called the FLEX*lm* client library) that provides the communication with the license server. On Windows, this module is called LMGR*xxx*.DLL, where *xxx* indicates the FLEX*lm* version. During execution, the application program communicates with the vendor daemon to request a license.

1.1.2 The License Request Process

When you run a "counted" FLEX*lm*-licensed application the following occurs:

1. The license module in the client application finds the license file, which includes the host name of the license server node and port number of the license manager daemon, *lmgrd*.
2. The client establishes a connection with the license manager daemon (*lmgrd*) and tells it what vendor daemon it needs to talk to.
3. *lmgrd* determines which machine and port correspond to the master vendor daemon and sends that information back to the client.
4. The client establishes a connection with the specified vendor daemon and sends its request for a license.
5. The vendor daemon checks in its memory to see if any licenses are available and sends a grant or denial back to the client.
6. The license module in the application grants or denies use of the feature, as appropriate.

"Uncounted" features (where the number of licenses is '0') do not require a server, and the FLEX*lm* client library routines in the application grant or deny usage based solely upon the license contents.

1.1.3 Configuring FLEX*lm*

Most of the parameters of FLEX*lm* are configurable by the license administrator. The license administrator can set:

- the location of the license file (though it's recommended that a copy or link of the license remains at the location where the application expects it)
- the location of all executables
- the location of all log files
- the TCP/IP port number used by the license manager process, *lmgrd*

In addition, the license administrator can reserve licenses for specific users, nodes, or groups, and control other license-related options. Changing parameters is discussed in Chapter 5, "The Options File" on page 29.

Note—Refer to your vendor's documentation before attempting to change file names, locations, or contents.

1.2 Getting Started Checklist

The following sections provide a quick overview of how to set up and use licensing for FLEX*lm*-licensed products. By scanning the list, you should be able to quickly find the areas of interest. Cross-references point to more details in other parts of this manual.

1.2.1 Installing Licensed Software

As a license administrator you are responsible for setting up licensing on your system or network. This section tells you how to do that. If you are an end-user of the application and you will not be involved in installing it, then go to Section 1.2.2, "Notes for End-Users," on page 5.

Remember that the installation guide for your application software is the final word on installing and configuring FLEX*lm*.

Generally, however, installing FLEX*lm* licensing requires the following steps:

1. Select your license server nodes and get their hostids. See Appendix A, "Hostids for FLEXlm-Supported Machines".
2. Give the hostids to your software vendor and get a license file (or the data to enter in the license file) in return.
3. Consider combining the new license file with any existing license files. See Chapter 3, "Multiple License Files" on page 19.
4. Determine if an options file is desired, and if so, set it up.
5. Determine where to install the FLEX*lm* utility programs such as *lmgrd*, and `lmutil` (`lmstat`/`lmdown`/etc.) and install them unless the your vendor's installation script does so for you.
6. Start *lmgrd* (the license daemon) manually; you may also want to set it up to start automatically at boot time. See Section 2.1.1, "Setting the Path to the License File at Start-Up (Unix)," on page 8.

These steps are discussed briefly below, with cross-references to the appropriate locations for more detail.

License Servers and Hostids

Before running any FLEX*lm*-licensed program using floating licenses, you will need to set up your license server node (or nodes). You must select which node or nodes to run your license servers on, and provide the hostid of those machines to your software vendor. For pointers on selecting your server machine, see Chapter 4, "Selecting Server Nodes" on page 23.

You can get the hostid of the server machine by running FLEX*lm*'s `lmhostid` utility on that machine. If you don't have `lmhostid`, you can get the hostid of your machine by using the appropriate command as described in Appendix A, "Hostids for FLEXlm-Supported Machines".

Using the hostid of your server machines your vendor will send you a license file that enables their application software.

LMGRD AND LICENSE FILES

Once you have received a license file from your vendor, you must install it on your system and start up the license manager daemon, *lmgrd*.

- Your software vendor may have selected a default location for your license file. If not, you can use any location you wish. For more details, see Chapter 2, "The License File" on page 7.
- Some vendors provide special scripts to start up the license daemon. If not, you can run *lmgrd* directly. To start *lmgrd* automatically at boot time, you will have to modify your system files (Unix), or use the FLEX*lm* Control Panel (Windows). For details see Section 2.1.1, "Setting the Path to the License File at Start-Up (Unix)," on page 8.

ADMINISTRATION TOOLS

GLOBE*trotter* Software supplies administration tools to your software vendor. The vendor usually includes them with their product. The recommended location for the tools is `/usr/local/bin` (Unix), `C:\flexlm` (Windows), or `SYS$COMMON:[SYSMGR]` (VMS), but you can install them in a different location (or not at all). See Chapter 6, "License Administration Tools" on page 39 for more information.

OPTIONS FILES

The options file controls various options such as reservations and timeouts of licenses. Most users run without an options file, but you may decide you want to use some options. For example, many administrators use an option to limit the quantity and content of logged messages. To set up an options file, see Chapter 5, "The Options File" on page 29.

1.2.2 Notes for End-Users

As a user of a FLEX*lm*-licensed application, you may need to know a few things to use the system effectively. The main things you need to know are:

- How to tell an application which license file to use.
- How to query the system to find out who is using a license.

HOW TO SPECIFY A LICENSE FILE

The license file determines what features are available to a program. It also contains information telling the application how to connect to the license server.

For information about the standard way of specifying a license file for an application, see Chapter 2, "The License File" on page 7.

GETTING INFORMATION ABOUT LICENSES

To find out who is using a license run `lmstat`, described in Chapter 6, "License Administration Tools" on page 39.

Chapter 2

The License File

The license file contains all site-specific information required by FLEX*lm*. This information includes:

- server names and hostids
- vendor names and paths to vendor daemon executables
- feature information

In general, the license file, or a copy of it, must be accessible to every machine that runs a FLEX*lm*-licensed application, and each machine designated as a license server. Before you can use the application you have to start the license manager daemon (*lmgrd*) using the following syntax:

```
lmgrd [-c license_file_path] (Unix and OS/2)
lmgrd -app [-c license_file_path] (Windows/NT)
```

where:	**is the:**
-c *license_file_path*	full pathname to the license file. Otherwise, LM_LICENSE_FILE is used, or the default location: `/usr/local/flexlm/licenses/license.dat` (Unix) or `C:\flexlm\license.dat` (Windows).

Note—On VMS and Netware systems there is no lmgrd and the vendor daemon is run directly.

2.1 Specifying Location of the License File

If your software vendor recommends a specific location for your license file, or if the default is not practical for you, use one of the following methods to put the license file in another location:

- set the path when you start *lmgrd* with "-c *path*"
- set the path with the LM_LICENSE_FILE environment variable

If you are running the application on multiple nodes, you have three options for making your license file available on all the machines:

- Place the license file in a partition which is available (via NFS on Unix systems) to all nodes in the network that need the license file.
- Copy the license file to all of the nodes where it is needed.
- Set `LM_LICENSE_FILE` to "*port@host*", where *host* and *port* come from the SERVER line in the license file.

Since the vendor daemon keeps track of license usage, and since the license file contains encrypted data to protect it against modification, you may move and copy the license file as much as necessary.

Note—You can only start *lmgrd* on the server node specified in the license file.

Note—If you are running redundant servers, you should have separate copies of the license file (as well as the binaries for *lmgrd* and the vendor daemons) on *each* server node. If you do not do this, you lose all the advantages of having redundant servers, since the file server holding these files becomes a single point of failure.

No matter which option you choose, you must first install *lmgrd* (on Unix and Windows/NT systems) and the vendor daemon.

2.1.1 Setting the Path to the License File at Start-Up (Unix)

Note—On Unix systems, since it is prudent to avoid using root for commands that do not require root permissions, (and no part of FLEX*lm* requires root permissions) it is strongly recommended that lmgrd be run as a non-privileged user (not "root").

To start the license manager daemon (*lmgrd*) execute a command similar to the following.

If you are running in the C shell:

```
% lmgrd_path -c license_file_path >& log_path &
```

If you are using either the Korn or Bourne shell:

```
$ nohup lmgrd_path -c license_file_path > log_path 2>&1 &
```

To start the *lmgrd* daemon automatically every time you reboot the license server add a line similar to the following to `/etc/rc.boot`, `/etc/rc.local`, `/etc/rc3.d/`*file* or the appropriate startup file:

[username's login shell is csh:]

```
su username -c "umask 022; lmgrd_path -c license_path >& log_path&"
```

[username's login shell is sh:]

```
su username -c "umask 022; lmgrd_path -c license_path > log_path 2>&1 &"
```

where:	**is the:**
username	non-privileged user (It is recommended that lmgrd NOT be run as root, for security reasons).
lmgrd_path	full pathname to the *lmgrd* executable.
license_path	full pathname to the license file.
log_path	full pathname to the debug log file.

Note—This will not start the daemon until you reboot your license server machine.

Because FLEX*lm* supports a large number of platforms, we can not describe all variations of system boot files in this manual. See your operating system's documentation for specific information about modifying startup files.

2.1.2 Setting the Path to the License File at Start-Up (Windows)

This is set with the FLEX*lm* Control Panel.

2.1.3 Setting the Path with an Environment Variable

Use the environment variable LM_LICENSE_FILE to set the location of the license file. For example in the C shell:

```
% setenv LM_LICENSE_FILE license_file_path
```

In the Korn and Bourne shells:

```
# LM_LICENSE_FILE=license_file_path
# export LM_LICENSE_FILE
```

On Windows 3.1 and 95, add the following line to C:\autoexec.bat:

```
SET LM_LICENSE_FILE=license_file_path
```

On NT, use the System Control Panel applet to change the global environment, adding LM_LICENSE_FILE to *license_file_path*

where:	is the:
license_file_path	full pathname to the license file. This can also be a *port@host* setting, where *port* and *host* are the port number and hostnames from the SERVER line in the license file.

Note—The "-c" option overrides the setting of the LM_LICENSE_FILE environment variable for *lmgrd* and other FLEX*lm* utilities like lmstat and lmdown. See Section 3.1.3, "Using Separate License Files on the Same Server Node," on page 21 for more information about LM_LICENSE_FILE.

Note—Some applications do not recognize the LM_LICENSE_FILE environment variable.

2.2 License File Format

License files usually begin with a SERVER line (or three lines for redundant servers) followed by one or more DAEMON lines, followed by one or more FEATURE or INCREMENT lines. In some cases the license file requires no SERVER line and no DAEMON line. See Section 4.4, "Counted vs. Uncounted Licenses," on page 27, for more information.

You can modify these data items in the license file:

- node names on the SERVER line(s)
- port numbers on SERVER line(s)
- pathnames on the DAEMON line(s)
- options file pathnames on DAEMON line(s)

- Optional port numbers on DAEMON line(s) (for firewall support only)
- USE_SERVER line (FLEX*lm* v5 and later only)
- values in "*name=value*" pairs on FEATURE lines, if *name* is all lowercase

Note—Everything else is used to compute the license key, and should be entered exactly as supplied by your software vendor. All data in the license file is case sensitive, unless otherwise indicated.

In the following sections, options modifiable by the license administrator are *italicized.*

2.2.1 SERVER Lines

The SERVER line specifies the node name and hostid of the license server, and the port number of the license manager daemon (*lmgrd*). Normally a license file has one SERVER line. Three SERVER lines mean that you are using redundant servers. The absence of a SERVER line means every FEATURE or INCREMENT line in the license file is uncounted. For more information about uncounted features, see Section 2.2.4, "FEATURE or INCREMENT Lines," on page 11. License administrators do not have the option of deleting SERVER lines from a license file because the hostids from the SERVER lines are encrypted into the passwords on every FEATURE and INCREMENT line. For more information about redundant servers, see Chapter 4, "Selecting Server Nodes" on page 23.

The format of the SERVER line is:

```
SERVER nodename id port-number
```

where:	**is the:**
nodename (or IP-address)	The system hostname. String returned by the UNIX *hostname* or *uname -n* command. On NT, *ipconfig /all*; on Win95, *winipcfg /all* return the hostname. If the application uses FLEX*lm* v5 or higher, this can be an IP-address (in *nnn.nnn.nnn.nnn* format).
id	Usually the string returned by the `lmhostid` command. This can only be changed by your software supplier.
port-number	TCP port number to use. A valid number is any unused port number between 0 and 64000. On Unix choose a port >1024, since <1024 are privileged port numbers.

Example:

```
SERVER enterprise 0122345 21987
```

2.2.2 DAEMON Lines

The DAEMON line specifies the daemon name and path. *lmgrd* uses this line to start the vendor daemons, and the vendor daemon reads it to find the options file. The format of the DAEMON line is shown below.

`DAEMON` *`daemon-name daemon_path`* `[[options=]`*`options_path`*`] [[port=]`*`portnum`*`]`

where:	is the:
daemon-name	name of the vendor daemon used to serve some feature(s) in the file. This name cannot be changed by the administrator.
path	pathname to the executable for this daemon. Generally the license administrator is free to install the daemon in any directory. (It is recommended however, that it be installed in a local directory on the license server node.)
options_path	full pathname of the end-user specified options file for this daemon. (See Chapter 5, "The Options File" on page 29.) FLEX*lm* does not require an options file. The keyword "options=" requires a v5+ vendor daemons.
portnum	Vendor daemon port number. NOTE: This is for firewall support only, and is otherwise not recommended. This requires a v5+ lmgrd.

Example:

```
DAEMON sampled /usr/local/sampled /usr/local/options/options.dat
```

Note—Versions of FLEX*lm* prior to v3.0 did not support the line continuation character ("\"). Previous versions required that each line in the license file fit on a single line.

2.2.3 USE_SERVER line (v5+ only)

USE_SERVER takes no arguments, and has no impact on the server. When the application sees USE_SERVER, it ignores everything in the license file except preceding SERVER lines, and the checkout validation occurs at the vendor daemon. USE_SERVER is recommended since it improves performance when a license server is used. For uncounted features, USE_SERVER can be used to force logging of usage by the daemons.

2.2.4 FEATURE or INCREMENT Lines

A FEATURE line describes the license to use a product. An INCREMENT line can be used in place of a FEATURE line, as well as to "incrementally" add licenses to a prior FEATURE or INCREMENT line in the license file.

Only the first FEATURE line for a given feature will be processed by the vendor daemon. If you want to have additional copies of the same feature (for example, to have multiple node locked counted features), then you must use multiple INCREMENT lines. INCREMENT lines form license groups based on the feature name, version, and node lock hostid. If the feature name, version, and node lock hostid (and optionally, the vendor string, if the vendor specified this) match a prior INCREMENT or FEATURE line, the new number of licenses is added to the old number. If any of the three do not match, a new group of licenses is created in the

vendor daemon, and this group is counted independently from others with the same feature name (called a "license pool"). INCREMENT is not available for pre-v2.61 FLEX*lm* clients or servers. A FEATURE line does not give an additional number of licenses, whereas an INCREMENT line ALWAYS gives an additional number of licenses.

Note—There is a rarely used option in FLEX*lm* which causes FEATURE lines to function as INCREMENT lines. This option is called `ls_use_all_feature_lines`. You will have to ask your vendor if they use this option. If they do, then all FEATURE lines behave exactly as INCREMENT lines.

Note—A FEATURE line placed after another FEATURE or INCREMENT line will be ignored, unless `ls_use_all_feature_lines` is set.

The format for the FEATURE line changed in FLEX*lm* v3.0. The old format is understood by new clients and servers, but the new format is more flexible.

Pre v3.0 format:

```
FEATURE|INCREMENT name vendor ver expdate #lic key "vendor_str" [hostid]
```

New format:

```
FEATURE|INCREMENT name vendor version exp_date #lic key \
[HOSTID=hostid] [VENDOR_STRING="vendor-string"] \
[vendor_info="..."] [dist_info="..."] [user_info="..."] \
[asset_info="..."] [ISSUER="..."] [NOTICE="..."] [ck=nnn] \
[OVERDRAFT=nnn] [DUP_GROUP=NONE|SITE|[UHDV]]
```

Note—Nothing in a FEATURE/INCREMENT line is editable, except for values in the "*name=value*" pairs where *name* is all lowercase.

where:	**is the:**
name	name given to the feature by the vendor.
vendor	name of the vendor daemon; also found in the DAEMON line. The specified daemon serves this feature.
version	version of this feature that is supported by this license.
exp_date	expiration date, for example, 7-may-1996. Note: If the year is 0 (or 00, 000, 0000) then the license never expires. Dates before 2000 can be 2- or 4-digit years. After 2000 they must be 4-digit years. The expiration date is fully year-2000 compliant.
#lic	number of concurrent licenses for this feature. If the number of users is set to 0, the licenses for this feature are uncounted and no *lmgrd* is required but a hostid on the FEATURE line is required. See Section 4.4, "Counted vs. Uncounted Licenses," on page 27.

key — license key for this FEATURE line. Two identical looking FEATURE or INCREMENT lines may have different license keys if the start dates are different.

The following fields are all optional (except for *vendor-string* in the old format). For optional fields of the "name=value" syntax, if the name is lowercase, it can be modified and the license will remain valid.

"vendor_string" — vendor-defined string, enclosed in double quotes. This string can contain any characters except a quote.

hostid — string returned by `lmhostid`. Used only if the feature is to be bound to a particular host, whether its use is counted or not. (Numeric hostids are case insensitive). See Appendix A, "Hostids for FLEXlm-Supported Machines" for more information.

DUP_GROUP=... — Duplicate Grouping parameter can be specified in the license in FLEX*lm* v4.0 and later. The syntax is:

```
DUP_GROUP=NONE|SITE|[UHDV]
        U = DUP_USER
        H = DUP_HOST
        D = DUP_DISPLAY
        V = DUP_VENDOR_DEF
```

Any combination of UHDV is allowed, and the DUP_MASK is the OR of the combination. For example "DUP_GROUP=UHD" means the duplicate grouping is (DUP_USERIDUP_HOSTIDUP_DISPLAY), so for a user on the same host and display, additional uses of a feature do not consume additional licenses.

HOSTID=*hostid* — Same as hostid above, but in the name=value pair syntax (FLEX*lm* v3.0 and later).

HOST_BASED[=*n*] — Hostnames must be specified in INCLUDE statements in the end-user options file, and the number of hosts is limited to *#lic*, or the number specified in "=*n*".

ISSUED=*dd-mmm-yyyy* — Date issued.

ISSUER="..." — Issuer of the license.

NOTICE="..." — A field for intellectual property notices.

OVERDRAFT=*nnn* — FLEX*lm* v4.0 and later. The `OVERDRAFT` policy allows your vendor to specify a number of additional licenses which users will be allowed to use, in addition to the licenses they have purchased. This allows your users to not

	be denied service when in a "temporary overdraft" state. Usage above the licensed limit will be reported by the FLEX*admin* reporting tool.
PLATFORMS="..."	Usage is limited to the listed platforms. (v5.11+ only).
SN=*serial_num*	Serial number, used to identify INCREMENT lines.
SUPERSEDE[="*f1 f2* ..."	If this appears, all licenses issued before the date specified in ISSUED= are *superseded* by this line, and become ineffective.
USER_BASED[=*n*]	Users must be specified in INCLUDE statements in the end-user options file, and the number of users are limited to *#lic*, or the number specified in "=*n*".
VENDOR_STRING="..."	Same as "vendor_string" above, but in *name=value* pair syntax.

Note—The following attributes can be changed or deleted by end-users. This is indicated by a lowercase *name*.

asset_info="..."	Additional information provided by the license administrator for asset management.
ck=*nnn*	A checksum, useful with the lmcksum utility, which will verify that the license has been entered correctly by the end-user.
dist_info="..."	Additional information provided by the software distributor.
user_info="..."	Additional information provided by the license administrator.
vendor_info="..."	Additional information provided by the software vendor.

Examples:

```
FEATURE xyz_app xyzd 2.300 31-dec-1997 20 1234567890 "xyz"
INCREMENT f1 demo 1.000 1-jan-0 5 12345678901234567890 \
              HOSTID=INTERNET=195.186.*.* NOTICE="Licensed to XYZ corp"
```

2.2.5 FEATURESET Lines

The FEATURESET line is used to prevent FEATURE lines from being added to or removed from the license file. The format of the FEATURESET line is shown below.

```
FEATURESET daemon-name key
```

Note—Nothing in a FEATURESET line can be edited. Use the FEATURESET line exactly as it comes from your vendor.

where:	**is the:**
daemon-name	name of the vendor daemon.

key	license key for this FEATURESET line. This key encrypts the keys of all features this daemon supports, so no FEATURE/INCREMENT lines can be removed, added, or rearranged in this license file.

Example:

```
FEATURESET sampled 12345678
```

where:	is the:
sampled	name of the vendor daemon
12345678	key generated by the vendor

2.2.6 PACKAGE Lines

The purpose of the PACKAGE line is to support two different licensing needs:

- to license a product SUITE, or
- to provide a more efficient way of distributing a license file that has a large number of features, which largely share the same FEATURE line arguments.

A PACKAGE line, by itself, does not license anything—it requires a matching FEATURE/INCREMENT line to license the whole PACKAGE. A PACKAGE line can be shipped by your software vendor with a product, independent of any licenses. Later, when you purchase a license for that package, one or more corresponding FEATURE/INCREMENT licenses will enable the PACKAGE.

Example:

```
PACKAGE pkg_name vendor pkg_version pkg_key COMPONENTS=pkg_list \
                        [ OPTIONS=pkg_options ]
```

where:	is the:
pkg_name	name of the PACKAGE. The corresponding FEATURE/INCREMENT line must have the same name.
vendor	name of the vendor daemon that supports this PACKAGE.
pkg_version	version of the PACKAGE. The enabling FEATURE/INCREMENT line must have the same version.
pkg_key	20-character license key.
pkg_list	list of components. The format is: `feature[:version[:num_lic]]` PACKAGEs must consist of at least one COMPONENT. Version and count are optional, and if left out, their values come from the corresponding FEATURE/INCREMENT line. *num_lic* is only legal if OPTIONS=SUITE is not set—in this case the resulting number of licenses will be

num_lic on the COMPONENTS line multiplied by the number of licenses in the FEATURE/INCREMENT line. Examples:

```
COMPONENTS="comp1 comp2 comp3 comp4"
COMPONENTS="comp1:1.5 comp2 comp3:2.0:4"
```

OPTIONS=SUITE — This is what distinguishes a suite PACKAGE from a PACKAGE used to ease distribution.

With OPTIONS=SUITE, the corresponding FEATURE of the same name as the package name is checked out in addition to the component feature being checked out.

If OPTIONS=SUITE is not set, then the corresponding FEATURE of the same name as the package is removed once the PACKAGE is enabled; and it is not checked out when a component feature is checked out.

Examples

```
PACKAGE suite demo 1.0 20CHARKEY \
                COMPONENTS="comp1 comp2" OPTIONS=SUITE
FEATURE suite demo 1.0 1-jan-0 5 20CHARKEY
```

This is a typical SUITE example. There are 2 features: comp1 and comp2, which are each version 1.0, with 5, non-expiring licenses available. When comp1 or comp2 are checked out, "suite" will also be checked out.

```
PACKAGE suite demo 1.0 20CHARKEY COMPONENTS="c1:1.5:2 c2:3.0:4"
FEATURE suite demo 1.0 1-jan-1999 3 20CHARKEY SN=123
```

In this example, the component versions override the FEATURE versions, and the number of licenses available for any component is the product of the 3 licenses for suite and the number of licenses for that component. The result is equivalent to:

```
FEATURE c1 demo 1.5 1-jan-1999 6 20CHARKEY SN=123
FEATURE c2 demo 3.0 1-jan-1999 12 20CHARKEY SN=123
```

2.2.7 UPGRADE Lines

```
UPGRADE name daemon fromversion version exp_date #lic key "string" \
            [hostid] ck=nnn
```

All the data is the same as for a FEATURE or INCREMENT line, with the addition of the *fromversion* field. An UPGRADE line removes up to the number of licenses specified from any old version (>= fromversion) and creates a new version with that same number of licenses.

For example, the two lines:

```
INCREMENT f1 demo 1.000 1-jan-1999 5 9BFAC03164EDB7BC0462 ""
UPGRADE f1 demo 1.000 2.000 1-jan-1999 2 1B9A30316207EC8CC0F7 ""
```

would result in 3 licenses of v1.0 of f1 and 2 licenses of v2.0 of f1.

UPGRADE will operate on the most recent FEATURE or INCREMENT line (i.e., closest preceding FEATURE or INCREMENT line) with a version number that is >= *fromversion*, and < *version*.

Note that UPGRADE does not work for node locked, uncounted licenses. A new FEATURE line should be issued in this case, since the license count is irrelevant.

2.3 Sample License File

This is an example of a license file for a single vendor with two features.

```
SERVER excellent_server 17007ea8 1700
DAEMON xyzd /etc/xyzd
FEATURE xyz_app1 xyzd 1.000 01-jan-1999 10 1EF890030EABF324 "*"
FEATURE xyz_app2 xyzd 1.000 01-jan-1999 10 0784561FE98BA073 "*"
```

The license file above would allow the license server `excellent_server` with the hostid `17007ea8` to serve 10 floating licenses for `xyz_app1` and `xyz_app2` to any user on the network.

2.4 Types of License Files

License files are created by the software vendor. License files can specify floating (concurrent) usage, node locked (both "counted" and "uncounted"), and any combination of floating, counted and uncounted.

2.4.1 Floating (Concurrent) Licenses

A *floating license* means anyone on the network can use the licensed software, up to the limit specified in the license file. (Also referred to as *concurrent usage* or *network licensing*.) Floating licenses have no hostids on the individual FEATURE lines. Floating licenses requires an *lmgrd* and a vendor daemon to be running to count the concurrent usage of the licenses.

An example of a license file that provides floating licenses is:

```
SERVER lulu 17001234 1700
DAEMON xyzd /etc/xyzd
FEATURE f1 xyzd 1.00 1-jan-99 2 key1 "*"
FEATURE f2 xyzd 1.00 1-jan-99 6 key2 "*"
FEATURE f3 xyzd 1.00 1-jan-99 1 key3 "*"
```

This license file specifies that two licenses for feature "f1", six licenses for feature "f2", and one license for feature "f3" are available anywhere on the network that can access the license server "lulu".

2.4.2 Node Locked Licenses

Node locking means the licensed software can only be used on one node. A node locked license has a hostid on any FEATURE line that is node locked to a particular host. There are two types of node locked licenses; *uncounted* and *counted.*

If the number of licenses is set to 0, then the license is *uncounted* and *unlimited* use is permitted on the specified node. This configuration does not require an *lmgrd* or a vendor daemon because it is not going to count the concurrent usage of the features.

The following license file allows unlimited usage of feature "f1" on the nodes with hostids of 12001234 and 1700ab12:

```
FEATURE f1 xyzd 1.000 1-jan-99 0 key1 "" 12001234
FEATURE f1 xyzd 1.000 1-jan-99 0 key2 "" 1700ab12
```

Alternately, in FLEX*lm* v5.0 or later, these 2 FEATURE lines could have been issued by your software vendor with a *hostid list*:

```
FEATURE f1 xyzd 1.000 1-jan-99 0 key HOSTID="12001234 1700ab12"
```

If these were the only FEATURE lines in this license file, no *lmgrd* daemon would be necessary and you should not start one.

The following license file allows three licenses for feature "f1" to be run, but only on the node with hostid 1300ab43. (In this case, the daemons should be run on the same node that runs the software, since there is no reason to run the daemons on another node.)

```
SERVER lulu 1300ab43 1700
DAEMON xyzd /etc/xyzy
FEATURE f1 zyzd 1.00 1-jan-99 3 key "" 1300ab43
```

2.4.3 Mixed Node Locked and Floating Licenses

Uncounted node locked and concurrent usage licenses can be mixed in the same license file.

The following license file allows unlimited use of feature "f1" on nodes 17001111 and 17002222, while allowing two other licenses for feature "f1" to be used anywhere else on the network:

```
SERVER lulu 17001234 1700
DAEMON xyzd C:\flexlm\xyzd.exe
FEATURE f1 xyzd 1.00 1-jan-1999 0 key1 "" 17001111
FEATURE f1 xyzd 1.00 1-jan-1999 0 key2 "" 17002222
FEATURE f1 xyzd 1.00 1-jan-1999 2 key3 ""
```

This configuration requires an *lmgrd* and a vendor daemon because the concurrent usage of the two licenses on the third FEATURE line are counted.

Chapter 3

Multiple License Files

Since more than 1000 vendors have chosen FLEX*lm* as their license manager, chances are good that you will have to administer licenses from more than one vendor or multiple products from the same vendor.

3.1 Overview of Combining License Files

When you are running FLEX*lm*-licensed products from multiple vendors, you may need to take steps to prevent licensing conflicts during installation. There are three ways you can accomplish this:

- Multiple license server nodes; each running one *lmgrd* and one license file
- One license server node running one *lmgrd* and one license file
- One license server node running multiple *lmgrd*s and multiple license files

Note that each *lmgrd* can only read a single license file. In the first option mentioned above, you will have more license servers to monitor; in the third option you have only one server but multiple *lmgrd*s to administer.

Your product's license file(s) define the license server(s) by hostname and hostid in the SERVER line(s) in the license file. If the license files for two or more products contain identical hostids on the SERVER line(s), then these files can be combined. If the license files for two products contain different hostids on a SERVER line, then the license servers for those products will be running on different nodes and the license files cannot be combined.

If you have two or more products whose license servers run on the same node (as specified by the SERVER lines in the license files), you may be able to combine the license files into a single license file. If the SERVER lines in those files have identical hostids, then you can combine the files into a single file. If the SERVER lines have different hostids, then you must keep the license files separate.

More precisely, you can combine two license files under the following conditions:

1. The number of SERVER lines in each file is the same.
2. The hostid field of each SERVER line in one file exactly matches the hostid field of each SERVER line in the other file.

Some possible reasons license files may not be compatible are:

- License files are set up to run on different server nodes, so hostids are different.

- One file is set up for single server (has only one SERVER line), the other is set up for redundant servers (has multiple SERVER lines).
- One vendor uses a custom hostid algorithm, so the hostids on the SERVER lines are different even though the files are for the same machine.

If your license files are compatible as described above, then you have the option of combining license files and running a single *lmgrd*, as described below in Section 3.1.1, "Combining License Files from Multiple Vendors," on page 20. If the license files are not compatible, then you must keep the license files separate and run separate copies of *lmgrd* for each license file, as described in Section 3.1.3, "Using Separate License Files on the Same Server Node," on page 21.

Note that you are not required to combine compatible license files; you always have the option of running separate *lmgrds*, and there is virtually no performance or system-load penalty for running separate lmgrd processes.

3.1.1 Combining License Files from Multiple Vendors

If your license files are compatible, you can combine them with any text editor. To combine license files, read all of the compatible license files into one file, then edit out the extra SERVER lines so that only one set of SERVER lines remains. Write out this data, and you have your combined license file.How about "If you combine license files from multiple vendors, it is a good idea to keep a copy of the combined license file in each vendor's default license file location. This way, your users can avoid having to set LM_LICENSE_FILE, because each package finds it's license information in the default place. On UNIX, you can do this with a symbolic link from each default location to the location of the combined license file.

3.1.2 FLEXlm Version Component Compatibility

When you combine license files for two different FLEX*lm*-licensed products, it may be the case that those products do not use the same version of FLEX*lm*. FLEX*lm* is designed to handle this situation. There are two basic compatibility rules for FLEX*lm*:

1. A newer *lmgrd* can be used with an older vendor daemon, but a newer vendor daemon might not work properly with an older *lmgrd*.
2. A newer vendor daemon (or *lmgrd*) can be used with an older client program, but a newer client program might not work properly with an older vendor daemon.

From these two compatibility rules come the simple rules for selecting which version of administration tools to use:

1. Always use the newest version of *lmgrd* and the newest version of each vendor daemon.
2. Use the newest FLEX*lm* utilities.

For specific application programs, you can use either the new or the old version (with the assumption that the vendor daemon that for that application is at least as new as the application).

3.1.3 Using Separate License Files on the Same Server Node

You must run a separate copy of *lmgrd* for each license file. When you run multiple copies of *lmgrd*, there are two details to remember:

1. The port number on the SERVER line of each license file must be unique. You can use a standard text editor to change the port number in each license file so that they are all different.
2. You must make sure that you are using a compatible version of *lmgrd* when you start it up for a particular license file. This can be done by using an explicit path to *lmgrd*.

When running client programs (such as a licensed application), you can set the LM_LICENSE_FILE environment variable to point to multiple license files. For example, you may have a license file from vendor ABC and a license file from vendor XYZ with incompatible servers. You can place the license file from vendor ABC into:

```
/usr/flexlm/abc.dat
```

and the license file from vendor XYZ into:

```
/usr/flexlm/xyz.dat
```

then set the LM_LICENSE_FILE environment variable to point to both of them. Each name in LM_LICENSE_FILE should be separated by a colon (":") on Unix systems, a semicolon (";") on Windows and Windows/NT systems (in FLEX*lm* v4.1, a comma was used on Windows and NT), and a space (" ") on VMS systems.
In the C shell:

```
% setenv LM_LICENSE_FILE /usr/flexlm/abc.dat:/usr/flexlm/xyz.dat
```

In the Korn and Bourne shells:

```
# LM_LICENSE_FILE=/usr/flexlm/abc.dat:/usr/flexlm/xyz.dat
# export LM_LICENSE_FILE
```

Note—LM_LICENSE_FILE can point to only one license file for FLEX*lm* v1.x applications.

3.2 Using LM_LICENSE_FILE License File List

If products use different license server nodes, each set of license servers requires separate license files. (When multiple software vendors use the same set of license server nodes, the technique described above in Section 3.1, "Overview of Combining License Files," on page 19 can be used to combine license files.) The resulting (multiple) license files can be installed in convenient locations. On Unix you would set the LM_LICENSE_FILE environment variable as follows:

```
% setenv LM_LICENSE_FILE lfpath1:lfpath2:....:lfpathN
```

Note—Use a colon (":") to separate the license file names on Unix, on Windows and Windows/NT use a semicolon (";"), and on VMS use a space (" ").

where:	**is the:**
lfpath1	path to the first license file
lfpath2	path to the second license file.
.	
.	
.	
lfpathN	path to the last (Nth) license file

Each application queries each license file in the order it is listed in the `LM_LICENSE_FILE` path. If the license server serving the license file listed in `lfpath1` is unreachable, perhaps due to an NFS problem, changing the `LM_LICENSE_FILE` allows a user to obtain a license from another server. *lfpath* can also be *"port@host"*, using the port-number and hostname from the SERVER line in the license file.

For more information about `LM_LICENSE_FILE` and setting the location of the license file, see Section 2.1, "Specifying Location of the License File," on page 7.

Chapter 4

Selecting Server Nodes

This chapter helps you decide which nodes to use as license server nodes.

4.1 Resources Used by the Server

This section discusses the resources used by the license server. When you select a server node, you may need to take into account the system limits on these resources. For small numbers of licenses (under about 100), most of these items should not be a problem on any workstation.

4.1.1 Sockets

When using TCP, a single vendor daemon can support as many users as the per-process system limit for file descriptors, which ranges from 256 on SunOS 4.x to 4000 on DEC Alpha. When no more file descriptors are available to a daemon, additional vendor daemons are spawned to allow for extra file descriptors, though this is not recommended. When using UDP, there is no limit to the number of end-users per vendor daemon process, since they can share a single socket in the vendor daemon (UDP has other drawbacks, and TCP is normally preferred). If there are more than 250 concurrent clients from a SunOS vendor daemon, it may be a good idea to move the server to a different OS, since all other OSs support more file descriptors. If there are more than 1000 concurrent clients being supported by a single vendor daemon, then it's probably good to split the license file into more than one file, from different servers, to lighten the networking traffic (which will require the ISV to agree to issue new licenses). Clients can checkout licenses from multiple servers using a license-file list via LM_LICENSE_FILE.

Each client connected to a license server uses one socket. The total number of sockets used by the license server programs is slightly larger than the total number of simultaneous clients.

On SCO systems, the default number of sockets may be set fairly low; if you choose to run a server on such a machine, you may need to reconfigure your kernel to have more sockets.

4.1.2 CPU Time

For small numbers of clients, the license servers use very little CPU time. The servers might have only a few seconds of CPU time after many days.

For a large number of clients (who are each exchanging heartbeat messages with the server), or for high checkout/checkin activity levels (hundreds per second), the amount of CPU time consumed by the server may start to become significant although, even here, CPU usage is normally not high. In this case, you may need to ensure that the server machine you select will have enough CPU cycles to spare.

Note—GLOBE*trotter* Software has rarely encountered a situation where CPU cycles were an issue, except where the vendor daemon ran out of sockets, and spawned another process.

4.1.3 Disk Space

The only output files created by the license servers are the debug and report log files. The report log files are used to generate accurate usage reports by FLEX*admin*. These log files contain one line for each checkout and one line for each checkin. If you have a lot of license activity, these log files will grow very large. You will need to consider where to put these files and how often to delete or prune them. The license administrator can opt not to log messages to the debug log file if disk space is at a premium. See Section 5.2.10, "NOLOG," on page 33 and Section 5.2.11, "REPORTLOG," on page 33.

Note that the log files should be local files on the server machine(s). See Section 4.2, "Diskless Nodes and Remote Mounted Disks," on page 25 for a discussion of remote file systems.

SWITCHING OUTPUT OF THE DEBUG LOG FILE ON UNIX SYSTEMS

The debug log file output can be switched after the daemons are running. The technique to do this involves piping the *stdout* of *lmgrd* to a shell script that appends to the file for each line.

This is done as follows:

Instead of the "normal" startup:

```
% lmgrd> LOG
```

Start *lmgrd* this way:

```
% lmgrd | sh -c 'while read line; do echo "$line" >> LOG ; done'
```

With this startup method, the output file "LOG" can be renamed and a new log file will be created. You could even make "LOG" a symbolic link and change the value of the link to switch the log file.

Note—This technique applies to Unix systems only.

Note—Since on Unix it is prudent to avoid running processes as root that do not require root permissions, we strongly recommend that you run lmgrd as a non-privileged user.

4.1.4 Memory

The FLEX*lm* daemons use little memory. On SunOS, *lmgrd* uses approximately100 kB and the vendor daemons use approximately 120 kB each.

4.1.5 Network Bandwidth

FLEX*lm* sends relatively small amounts of data across the network. Each transaction, such as a checkout or checkin, is typically satisfied with less than 1Kbyte of data transferred. This means that FLEX*lm* licensing can be effectively run over slow networks (such as dial-up SLIP lines) for small numbers of clients.

For a large number of clients (hundreds), each of which will be exchanging heartbeat messages with the vendor daemon, the network bandwidth used may start to become significant. In this case you should run client and server on the same local area network, which may require splitting licenses between 2 files for 2 servers. Users can use a license file list in LM_LICENSE_FILE to have effective access to both servers.

In high-traffic networks, with FLEX*lm* clients older than v5, you may also want to avoid setting `LM_LICENSE_FILE` to a `port@host` address. Instead, the license administrator should place a copy of the license file in a filesystem local to the application. See Section 2.1, "Specifying Location of the License File," on page 7.

4.2 Diskless Nodes and Remote Mounted Disks

GLOBE*trotter* Software recommends that you do not use remote mounted disks when you run the license server. In other words, we recommend that *lmgrd*, the vendor daemons, the license file, and the debug and report log files are all on locally mounted disks. If any of these files is on a remote mounted disk, you double the points of failure which could lead to a loss of all of your licenses. When all files are mounted locally, the licenses will be available as long as the server machine is up; but when the files are on a different machine, then the loss of either the license server machine or the file server machine will cause the licenses to be unavailable.

Diskless nodes are the extreme case of remote disks. We recommend that you do not use diskless nodes as license servers, since the files are necessarily accessed from a remote disk. In addition, FLEX*lm* sometimes (at the option of the vendor) makes a security check which fails on a diskless node. If you find that you are having problems with a lock file, one possibility is that you are attempting to run on a diskless node. The simplest solution is to select a different node for your license server.

4.3 Redundant Servers

FLEX*lm* supports two methods of redundancy: A set of three redundant license servers, and redundancy via a license file list in the $LM_LICENSE_FILE setting.

With three-server redundancy, if any two of the three license servers are up and running, the system is functional and hands out its total complement of licenses (Two out of three license servers is referred to as a "quorum").

With $LM_LICENSE_FILE list redundancy, each one of a group of license servers serves a subset of the total licenses. The end-user sets LM_LICENSE_FILE to a list of license files, where each license file refers to one of the license servers. The application then tries each server in the list, in order, until it succeeds or gets to the end of the list.

4.3.1 Three Server redundancy

SELECTING SERVER NODES

If all the end-user data is on a single file server, then there is no need for redundant servers, and GLOBE*trotter* Software recommends the use of a single server node for the FLEX*lm* daemons. If the end-user's data is split among two or more server nodes and work is still possible when one of these nodes goes down or off the network, then multiple server nodes can be employed. In all cases, an effort should be made to select stable systems as server nodes; in other words, do not pick systems that are frequently rebooted or shut down for one reason or another. The three server nodes can be any supported server nodes — it is not required that they be the same architecture or operating system.

These three-server redundant servers should have excellent communications. This form of redundancy requires that the servers exchange heartbeats periodically, and poor communications can cause poor performance. You should never configure redundant servers with slow communications or dialup links.

4.3.2 Redundancy via License File List in $LM_LICENSE_FILE

This is best explained by example. If 10 licenses are desired for both f1 and f2, the ISV would issue 2 sets of licenses with a count of 5 for each of f1 and f2. The server nodes (unlike three-server redundancy) can be physically distant. The license files would look like:

- License 1 for chicago

```
SERVER chicago 17007ea8 1700
DAEMON demo /etc/mydaemon
FEATURE f1 demo 1.000 01-jan-99 5 26C7DD9CD665B8270186 ""
FEATURE f2 demo 1.000 01-jan-99 5 0739D2F78CE46C57041D ""
```

- License 2 for tokyo

```
SERVER tokyo 17007ea8 1700
DAEMON demo /etc/mydaemon
FEATURE f1 demo 1.000 01-jan-99 5 16BE40E1DAEEEDA8798D ""
FEATURE f2 demo 1.000 01-jan-99 5 6DB6F3E40E61885712DF ""
```

The user in Chicago could set $LM_LICENSE_FILE to

```
1700@chicago:1700@tokyo
```

the user in Tokyo could set $LM_LICENSE_FILE to

```
1700@tokyo:1700@chicago
```

The application attempts the first server in the list, and if that fails for any reason, the second server is tried.

4.3.3 Comparing Three Server to License File List

ARE THERE ANY DRAWBACKS TO USING THE LICENSE FILE LIST FOR REDUNDANCY?

Yes. By default, once a *license job* has successfully checked out a license from one host, all subsequent checkouts must be satisfied from the same host. If the application requires more than one FEATURE, this could result in a license denial when the license is available on another server. An application can bypass this restriction if it is coded with the use of multiple FLEX*lm license jobs*. Only your application vendor can tell you if their application is programmed in this manner.

If the application supports license queueing, all licenses are only queued from the first host on the list.

Finally, if one server becomes unavailable, some licenses will be unavailable.

WHEN IS IT RECOMMENDED TO USE A LICENSE FILE LIST FOR REDUNDANCY RATHER THAN TRUE REDUNDANT SERVERS?

When there's less system administration available to monitor license servers, and when the applications are not mission-critical. The license file list has some other advantages: it's more forgiving if you lose quorum; it's not limited to 3 servers (any number will work); and for wide-area networks, you can make servers available locally, with remote servers available as backup.

4.4 Counted vs. Uncounted Licenses

The license file determines whether a license server is needed. If all the FEATURE (or INCREMENT) lines have a license-count of 0 (UNLIMITED), then no server is needed. This type of license is called *uncounted*. Alternatively, if any FEATURE lines have a non-zero license-count, then a server is required to count those licenses. If a vendor wants to use FLEX*lm* without a server, they must issue *uncounted* licenses.

With FLEX*lm* v5 or later, the license server can serve uncounted licenses as well. This is done so that the REPORTLOG file will include transactions for all license requests, which can then be reported on by FLEX*admin*. To do this, include a SERVER line in the license file, and put the USE_SERVER line immediately after the SERVER line. The vendor daemon will service the uncounted licenses, and the USE_SERVER line indicates to applications that they will be authorized by the server.

4 *Selecting Server Nodes*

Chapter 5

The Options File

The options file allows the license administrator to control various operating parameters of FLEX*lm*. Specifically the license administrator can:

- Allow the use of features based on user, hostname, or display name.
- Deny the use of features based on user, hostname, or display name.
- Reserve licenses based on user, hostname, or display name.
- Control the amount of information logged about license usage.

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

Note—Lines in the options file were limited to 200 characters prior to FLEX*lm* v3.0. In v3.0 and later, the line length is 2048 characters. FLEX*lm* v4.0 allows the "\" character as a continuation character in options file lines.

5.1 Creating an Options File

To create an options file:

1. Use the appropriate options listed in Section 5.2, "Customizing the Options File," on page 29 to create the options file using any text editor. You can put the options file anywhere; however, we recommend that the options file for vendor *xyz* be placed in

   ```
   /usr/local/flexlm/options/xyz.opt (Unix)
   C:\flexlm\options\xyz.opt (Windows/NT)
   ```

2. Add the pathname to the options file in the license file as the fourth field on the DAEMON line for the application's vendor daemon. For example,

   ```
   DAEMON xyzd /etc/xyzd /usr/local/flexlm/options/xyz.opt
   ```

   would enable the xyzd vendor daemon to look at the specified options file.

5.2 Customizing the Options File

Below is an overview of the options file syntax. See Section 5.4, "Understanding Options Files," on page 35 for examples and additional information.

Each line of the file controls one option. The options are:

EXCLUDE	deny a user access to a feature.
EXCLUDEALL	deny a user access to *all* features served by this vendor daemon.

GROUP	define a group of users for use with any options.
HOST_GROUP	define a group of hosts for use with any options.
INCLUDE	allow a user to use a feature.
INCLUDEALL	allow a user to use *all* features served by this vendor daemon.
LINGER	cause licenses to be held by the vendor daemon for a period after the application checks them in or exits.
MAX	limit usage for a particular feature/group—prioritizes usage among users.
MAX_OVERDRAFT	limit overdraft usage below the amount enabled in the license.
NOLOG	turn off logging certain items.
REPORTLOG	specify that a logfile be written suitable for use by the FLEX*admin* End-User Administration Tool.
RESERVE	reserve licenses for a user.
TIMEOUT	specify idle timeout for a feature, returning it to the free pool for use by another user.
TIMEOUTALL	Set timeout on all features.

You can include comments in your options file by starting each comment line with a pound sign "#". Everything in an options file is case sensitive. Be sure that user names and feature names, for example, are entered correctly.

FEATURENAME SPECIFICATION

In FLEX*lm* v5, you can select a particular line of a given featurename, as follows:

```
featurename:name=value
```

For example:

```
f1:VERSION=2.0
```

You can specify a feature by any of the following fields:

```
VERSION HOSTID EXPDATE KEY VENDOR_STRING ISSUER NOTICE dist_info
user_info asset_info
```

In FLEX*lm* v5.11 or later, you can use a PACKAGE name in place of a feature name, and the option will apply to the whole package.

5.2.1 EXCLUDE

```
EXCLUDE featurename type name
```

Excludes a user, host, display, or group from the list of who is allowed to use the feature. Excluded users will not be allowed to use the feature.

- *featurename* - name of the feature being affected

- *type* - one of USER, HOST, DISPLAY, GROUP or HOST_GROUP (see Section 5.2.3, "GROUP" and Section 5.2.4, "HOST_GROUP," on page 31
- *name* - name of the user or group to exclude

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

```
EXCLUDE f1 USER hank
```

5.2.2 EXCLUDEALL

`EXCLUDEALL` *`type name`*

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

- *type* - one of USER, HOST, DISPLAY, GROUP or HOST_GROUP (see Section 5.2.3, "GROUP" and Section 5.2.4, "HOST_GROUP," on page 31
- *name* - name of the user or group to exclude

To exclude any user on the server "chaos" from using all features served by this vendor daemon:

```
EXCLUDEALL HOST chaos
```

5.2.3 GROUP

`GROUP` *`groupname usernamelist`*

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

- *groupname* - name of the group being defined
- *usernamelist* - list of user names in that group

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

```
GROUP Hackers bob howard james
```

Note—In FLEX*lm* v3.0 multiple GROUP lines will add all the users specified into the group. Pre-v3.0 FLEX*lm* daemons do not allow multiple GROUP lines to concatenate. In fact, the second GROUP line would re-define the GROUP.

Note—In FLEX*lm* v4.0 or later, USER_GROUP is an alias for GROUP.

5.2.4 HOST_GROUP

`HOST_GROUP` *`groupname hostnamelist`*

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

- *groupname* - name of the group being defined
- *hostnamelist* - list of host names in that group

To define the group accounting consisting of node_a, node_b, and node_c:

```
HOST_GROUP accounting node_a node_b node_c
```

Note—HOST_GROUP is available for FLEX*lm* v4.0 and later.

5.2.5 INCLUDE

`INCLUDE` *`featurename type name`*

Includes a user, host, display, or group in the list of who is allowed to use the feature. Anyone not in an INCLUDE statement will not be allowed to use that feature.

- *featurename* - name of the feature being affected
- *type* - one of USER, HOST, DISPLAY, GROUP, or HOST_GROUP (see Section 5.2.3, "GROUP" and Section 5.2.4, "HOST_GROUP," on page 31
- *name* - name of the user or group to include

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

```
INCLUDE f1 USER bob
```

Note—INCLUDE is required for USER_BASED features. The system administrator specifies which users are allowed to use the product, via INCLUDE, and the license limits the number of users that can be INCLUDEd.

5.2.6 INCLUDEALL

`INCLUDEALL` *`type name`*

Includes a user, host, display, or group in the list of who is allowed to use all features served by this vendor daemon. Anyone not in an INCLUDEALL statement will not be allowed to use these features.

- *type* - one of USER, HOST, DISPLAY, GROUP, or HOST_GROUP (see Section 5.2.3, "GROUP" and Section 5.2.4, "HOST_GROUP," on page 31
- *name* - name of the user or group to include

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

```
INCLUDEALL USER jane
```

5.2.7 LINGER

`LINGER` *`featurename interval`*

Rarely used. This causes the daemon to "hold on" to the license for *featurename* for *interval* seconds after the application checks the license in or exits. This could be useful for short-duration programs which will be used many times in a row by the same user, to ensure that the user will be able to re-acquire the license repeatedly. On the other hand, other users have to wait until the first user is completely finished, plus a linger interval. This is only useful if the application uses duplicate grouping. Otherwise, LINGER will cause you to use extra licenses. Contact your software vendor for information about how they implemented duplicate grouping in their product.

5.2.8 MAX

(v5.11+ vendor daemon only).

`MAX numlic featurename type name`

Limits usage for a group or user.

- *numlic* - usage limit for this user or group
- *featurename* - feature this limit applies to
- *type* - USER, HOST, DISPLAY, GROUP, or HOST_GROUP.
- *name* - name of the user or group to limit

5.2.9 MAX_OVERDRAFT

`MAX_OVERDRAFT featurename numlic`

Limits usage below the OVERDRAFT allowed by the license file.

5.2.10 NOLOG

`NOLOG what`

Turns off logging of specific events by the FLEX*lm* daemons.

- *what* - what to turn off; one of IN, OUT, DENIED, or QUEUED

To turn off logging of checkins:

```
NOLOG IN
```

To turn off logging of checkouts and queued requests two separate NOLOG lines are required:

```
NOLOG DENIED
NOLOG QUEUED
```

Note—License administrators might use this option to reduce the size of the *lmgrd* or "debug" log file.

5.2.11 REPORTLOG

`REPORTLOG filename`

REPORTLOG specifies the file which will contain the report-writer log for this vendor daemon. If *filename* begins with a '+' character, the file will be opened for append, otherwise the file will be overwritten each time the daemon is started. *FLEXadmin*™, a separate product available from Globetrotter, can be used to read and report on REPORTLOG files.

Note—This file is only useful with the FLEX*admin* license administration utility.

5.2.12 RESERVE

`RESERVE numlic featurename type name`

Reserves licenses for a specific user.

- *numlic* - number of licenses to reserve
- *featurename* - name of feature to reserve
- *type* - one of USER, HOST, DISPLAY, GROUP, or HOST_GROUP (see Section 5.2.3, "GROUP" and Section 5.2.4, "HOST_GROUP," on page 31
- *name* - name of the user or group to reserve licenses for.

To reserve one license of feature f1 for user "mel":

```
RESERVE 1 f1 USER mel
```

Note—Any licenses reserved for a user are dedicated to that user. Even when that user is not actively using the license it will be unavailable to other users. However, a RESERVED license will not cause an overdraft to be reported by FLEX*admin* if the license is not actually in use.

5.2.13 TIMEOUT

`TIMEOUT featurename seconds`

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

- *featurename* - name of the feature
- *seconds*- number of seconds after which inactive license is reclaimed

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

```
TIMEOUT f1 3600
```

TIMEOUT checks in the licenses if the process has been "idle" for a period longer than the specified time period, and someone else wants the license. The daemon declares a process idle when it has not heard from the process (the client sends heartbeats). The application must explicitly declare itself idle for this to work, or (on Unix) the application must be stopped (^Z). That is, unless the application explicitly supports this feature, it will not work. Contact your software vendor for information about how they implemented this feature in their product.

The application vendor can also disable the timeout feature, in which case the TIMEOUT option has no effect. The vendor can set a minimum value for the timeout. If you specify a timeout value smaller than the minimum, the minimum is used. The default minimum value is 900 seconds (15 minutes).

If you do not specify a timeout value in your options file, then there will be no timeout for that feature. Licenses are only freed by TIMEOUT when a new request for a license is made.

5.3 TIMEOUTALL

(v5.11+ vendor daemon only)

`TIMEOUTALL seconds`

Same as TIMEOUT, but applies to all features.

5.4 Understanding Options Files

The following information gives an overview of the syntax of a complete options file and some samples intended to illustrate ways to effectively control access to your licenses.

5.4.1 How the Vendor Daemon Uses the Options File

When the vendor daemon is started by *lmgrd*, it is passed the location of the options file. The location is specified in the license file for that product, on the DAEMON line. If no file is listed the daemon will not use any options file.

There can only be one options file per vendor daemon and each vendor needs a separate options file.

5.4.2 Rules of Precedence in Options Files

Before you can use options to utilize licenses effectively you must understand the options file precedence. INCLUDE and EXCLUDE statements can be combined in the same options file and control access to the same features. When doing so, keep in mind the following:

- If there is only an EXCLUDE list, everyone who is not on the list will be allowed to use the feature.
- If there is only an INCLUDE list, only those users on the list will be allowed to use the feature.
- If neither list exists, then everyone is allowed to use the feature.
- The EXCLUDE list is checked before the INCLUDE list; so someone who is on both lists will not be 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 an 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 can either:

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

5.4.3 Simple Options File Example

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

This options file would:

- Reserve one license for the feature "compile" for the user "robert."

- Reserve three licenses for the feature "compile" for anyone on a computer with the hostname "mainline."
- Prevent the user "lori" from using the "compile" feature on any node on the network.
- Cause 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 `/usr/local/flexlm/options/local.options`, then you would modify the license file DAEMON line as follows:

```
DAEMON xyzd /usr/local/xyzd /usr/local/flexlm/options/local.options
```

5.4.4 Limiting Access for Multiple Users

Each INCLUDE, INCLUDEALL, EXCLUDE, EXCLUDEALL, 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" we could create the following options file:

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

There is an easier way though. Create a GROUP and exclude the list of users from using the feature. Like the previous example, the following options file would exclude "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.

The GROUP function works for a list of user names prior to FLEX*lm* v4.0. To control access to multiple displays (and hosts in pre-v4.0 FLEX*lm*) you must use multiple option lines in your options file. For example, in pre-v4.0 FLEX*lm* 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:

```
EXCLUDE f1 USER fred
EXCLUDE f1 USER barney
```

In FLEX*lm* v4.0, you can 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
```

Note—See Section 5.2.3, "GROUP" and Section 5.2.4, "HOST_GROUP," on page 31 for more information about defining groups.

5.4.5 EXCLUDE Example

```
#First Define the group "painters"
GROUP painters picasso mondrian klee
EXCLUDE spell GROUP painters
EXCLUDE spell USER bob
EXCLUDE spell HOST bigbrush
```

This options file would:

- Prevent the users "picasso," "mondrian," and "klee" from using the feature "spell" on any machine on the network.
- Prevent the user "bob" from using the feature "spell" on any machine on the network.
- Prevent any user logged into the host "bigbrush" from using the feature "spell"
- Allow any other user, as long as they are not on "bigbrush", *and* they are not in "painters" *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.

5.4.6 INCLUDE Example

```
INCLUDE paint USER picasso
INCLUDE paint USER mondrain
INCLUDE paint HOST bigbrush
```

This options file would:

- Allow the user "picasso" to use the feature "paint" on any machine on the network.
- Allow the user "mondrain" to use the feature "paint" on any machine on the network.
- Allow any user, as long as they are on the host "bigbrush", to use feature "paint"
- Deny access to the feature "paint" to anyone except "picasso", "mondrain" or anyone from the host "bigbrush" (By implication.)

5 *The Options File*

Chapter 6

License Administration Tools

FLEX*lm* provides utilities for the license administrator to help manage the licensing activities on the network. These utilities are:

- *lmcksum* (v2.4 or later) - prints license checksums. (*page 40*)
- *lmdiag* (v4.0 or later) - diagnoses license checkout problems. (*page 40*)
- *lmdown* - gracefully shuts down all license daemons (both *lmgrd* and all vendor daemons) on the license server node (or on all three nodes in the case of redundant servers). (*page 41*)
- *lmgrd* - the main daemon program for FLEX*lm*. (*page 41*)
- *lmhostid* - reports the hostid of a system. (*page 42*)
- *lmremove* - removes a single user's license for a specified feature. (*page 43*)
- *lmreread* - causes the license daemon to reread the license file and start any new vendor daemons. (*page 44*)
- *lmstat* - helps you monitor the status of all network licensing activities. (*page 44*)
- *lmswitch* (VMS only) -switches the debug logfile. (*page 44*)
- *lmswitchr* - switches the report log file. (*page 45*)
- *lmver* - reports the FLEX*lm* version of a library or binary file. (*page 45*)

Beginning in FLEX*lm* v2.4, all FLEX*lm* utility programs (except *lmgrd*) are packaged as a single executable called *lmutil*. lmutil can either be installed as the individual commands (either by creating links to the individual command names, or making copies of lmutil as the individual command names), or the commands can be run as "lmutil command", e.g. "lmutil lmstat", or "lmutil lmdown". On Windows or Windows/NT systems, the "lmutil command_name" form of the commands are available. There is also a Windows version of these commands - see Section 6.12.1, "License Administration Tools—LMTOOLS for Windows," on page 46.

Most *lmutil* utilities need to know the path to the license file. This can be specified with a "-c *license_file*" argument, or by setting the LM_LICENSE_FILE environment variable. Otherwise, the default location is used.

6.1 lmcksum

The `lmcksum` program (FLEX*lm* v2.4 or later) will perform a checksum of a license file. This is useful to verify data entry errors at your location. `lmcksum` will print a line-by-line checksum for the file as well as an overall file checksum. `lmcksum` takes the "-k" switch to force the encryption key checksum to be case-sensitive.

`lmcksum` will ignore all fields that do not enter into the encryption key computation; thus the server node name and port number, as well as the daemon pathname and options file names are not checksummed. In addition, `lmcksum` will treat non-case sensitive fields correctly (in general, `lmcksum` is not case-sensitive).

`lmcksum` takes an optional daemon name; if specified, only license file lines for the selected daemon are used to compute the checksums.

For FEATURE lines that contain ck=nnn, lmcksum prints simply OK or BAD. This is available for FLEX*lm* products v4 or higher.

Usage is:

`lmcksum [-c license_file]`

where:	**is the:**
-c *license_file*	path to the file to checksum. By default lmcksum uses "license.dat" in the current directory (unlike other *lmutil* commands).

Example output is:

```
lmcksum - Copyright (C) 1989, 1997 GLOBEtrotter Software, Inc.
lmcksum: using license file "/usr/local/flexlm/licenses/license.dat"

       189: SERVER speedy 08002b32b161 2837
       166: DAEMON demo C:\flexlm\demo.exe
         8: FEATURE f1 demo 1.000 01-jan-99 0 3B2BC33CE4E1B8F3A0BF ""
OK:    231: FEATURE f2demo 1.0 01-jan-0 1 8B1C30015351B7737F5E \
              DUP_GROUP=HD ck=231
       109: (overall file checksum)
```

6.2 lmdiag

`lmdiag` (FLEX*lm* v4.0 or later) allows you to diagnose problems when you cannot check out a license.

Usage is:

`lmdiag [-c license_file] [-n] [feature]`

where:	**is the:**
-c *license_file*	path to the file to diagnose.

-n	run in non-interactive mode; `lmdiag` will not prompt for any input in this mode. In this mode, extended connection diagnostics are not available.
feature	diagnose this feature only.

If no *`feature`* is specified, `lmdiag` will operate on all features in the license file(s) in your path. `lmdiag` will first print information about the license, then attempt to check out each license. If the checkout succeeds, `lmdiag` will indicate this. If the checkout fails, `lmdiag` will give you the reason for the failure. If the checkout fails because `lmdiag` cannot connect to the license server, then you have the option of running "extended connection diagnostics".

These extended diagnostics attempt to connect to each port on the license server node, and can detect if the port number in the license file is incorrect. `lmdiag` will indicate each port number that is listening, and if it is an *lmgrd* process, `lmdiag` will indicate this as well. If `lmdiag` finds the vendor daemon for the feature being tested, then it will indicate the correct port number for the license file to correct the problem.

See Also: Appendix B, "FLEXLM_DIAGNOSTICS" on page 54.

6.3 lmdown

The `lmdown` utility allows for the graceful shutdown of all license daemons (both *lmgrd* and all vendor daemons) on all nodes.

Usage is:

```
lmdown [-c license_file] [-q]
```

where:	**is:**
-c *license_file*	Use the specified license file.
-q	Don't prompt or print a header. Otherwise `lmdown` asks "Are you sure? [y/n]: ".

You should protect the execution of `lmdown`, since shutting down the servers causes users to lose their licenses. See the "`-p`" or the "-x" options in Section 6.4, "lmgrd," on page 41 for details about securing access to `lmdown`.

Note—When shutting down redundant servers, there is a 1 minute delay before the servers shut down. Do **not** use "kill -9" to shut down the license servers.

6.4 lmgrd

`lmgrd` is the main daemon program for FLEX*lm*. When you invoke *lmgrd*, it looks for the license file which contains the information about vendors and features. On Unix systems, it is strongly recommended that lmgrd be run as a non-privileged user (not *root*).

Usage is:

```
lmgrd [ -app ] [ -c license_file ] [ -t timeout_interval ] [ -l logfile ]
[ -s timestamp_interval ] [ -2 -p ] [ -v ] [ -x lmdown ] [ -x lmremove ]
```

where:	is:
-app	Required for Windows/NT systems.
-c *license_file*	Use the license file named.
-t *timeout_interval*	Sets a timeout interval, in seconds, during which redundant daemons must complete their connections to each other. The default value is 10 seconds. A larger value may be desirable if the daemons are being run on busy systems or a very heavily loaded network.
-l *logfile*	Write the debug log to *logfile*.
-s *timestamp_interval*	Specifies the logfile timestamp interval, in minutes. The default is 360 minutes.
-2 -p	Restricts usage of `lmdown`, `lmreread`, and `lmremove` to a FLEX*lm* administrator who is by default root. If there a UNIX group called "lmadmin" then use is restricted to only members of that group. If root is not a member of this group, then root does not have permission to use any of the above utilities. The "`-p`" option is available in FLEX*lm* v2.4 and later.
-v	Prints *lmgrd*'s version number and copyright and exits.
-x lmdown	Disallow the lmdown command (no user can run lmdown). If lmdown is disabled, you will need to stop lmgrd via "`kill` *pid*" (Unix) or CTRL-ALT-DEL and stop the lmgrd and vendor daemon processes (Windows 95). On Unix, be sure the kill command does not have a -9 argument.
-x lmremove	Disallow the lmremove command (no user can run lmremove)

Note—The -x lmdown and -x lmremove options are available in FLEX*lm* v4.0 and later.

6.5 lmhostid

The `lmhostid` utility reports the hostid of a system.

Usage is:

```
lmhostid [-n]
```

The output of this command looks as follows:

```
lmhostid - Copyright (c) 1989, 1990 Globetrotter Software, Inc.
The FLEXlm hostid of this machine is "69021c89"
```

With the "-n" argument, no header is printed; only the hostid.

See Appendix A, "Hostids for FLEXlm-Supported Machines".

6.6 lmremove

The `lmremove` utility allows you to remove a single user's license for a specified feature. This is only needed when a client node crashes, since that's the only condition where a license is not automatically freed. If the application is active, it will re-checkout the license after it is freed by lmremove.

Usage is:

```
lmremove [ -c file ] feature user host display
       or
lmremove [ -c file ] -h feature host port handle
```

where:	**is the:**
-c *license_file*	license file
feature	name of the feature checked out by the user.
user	name of the user whose license you are removing (from lmstat -a).
host	name of the host the user is logged in to (from lmstat -a).
display	name of the display where the user is working (from lmstat -a).
port	port, as reported by lmstat -a
handle	handle, as reported by lmstat -a

The *user host display port and handle* information must be obtained from the output of `lmstat -a`.

lmremove removes all instances of *user* on *host* on *display* from usage of *feature*. If the optional "-c *file*" is specified, the indicated file is used as the license file. You should protect the execution of `lmremove`, since removing a user's license can be disruptive. See the "`-p`" or the "-x" options in Section 6.4, "lmgrd," on page 41 for details about securing access to `lmremove`.

The `-h` variation uses the serverhost, port, and license handle, as reported by `lmstat -a`. Consider this example `lmstat -a` output:

```
joe cloud7 /dev/ttyp5 (v1.000) (cloud9/7654 102), start Fri 10/29 18:40
```

In this example, the serverhost is "cloud9", the port is "7654" and the license handle is 102. To remove this license, issue the following command:

```
lmremove -h f1 cloud9 7654 102
```

or

```
lmremove f1 joe cloud7 /dev/ttyp5
```

When removing by handle, if licenses are grouped as duplicates, all duplicate licenses will also be removed.

6.7 lmreread

The `lmreread` utility causes the license daemon to reread the license file and start any new vendor daemons that have been added. In addition, all pre-existing daemons will be signaled to reread the license file for changes in feature licensing information.

Usage is:

```
lmreread [-c license_file]
```

The license administrator may want to protect the execution of `lmreread`, since removing a user's license can be disruptive. See the "-p" and "-x" options in Section 6.4, "lmgrd," on page 41 for details about securing access to `lmreread`.

Note—If you use the "`-c`" option, the license file specified will be read by `lmreread`, not by *lmgrd*; *lmgrd* rereads the file it read originally. Also, `lmreread` cannot be used to change server node names or port numbers. Vendor daemons will not reread their option files as a result of `lmreread`.

6.8 lmstat

The `lmstat` utility helps you monitor the status of all network licensing activities.

Usage is:

```
lmstat [-a] [ -A ] [-c license_file] [-f feature] [-i [feature]]
       [-S vendor] [-s hostname] [-t value]
```

where:	**is:**
-a	Display all information
-A	List all active licenses
-c *license_file*	Use the license file named.
-f *feature_name*	List users of *feature_name*.
-i [*feature_name*]	Print information about the named *feature*, or all features if no *feature_name* is given. This option is usually not recommended, since the information does not come from the license server, and may not reflect what the server actually supports.
-S [*vendor*]	List all users of *vendor*'s features.
-s *hostname*	Display status of clients running on *hostname*.
-t *value*	Set `lmstat` timeout to "value".

6.9 lmswitch

Note—The `lmswitch` command is available on VMS only.

The `lmswitch` utility switches the debug log file for the daemon serving the specified feature while the daemon is running.

Usage is:

`lmswitch` *`feature new-file`*

where:	**is the:**
feature	any feature this daemon supports.
new-file	the new file path.

Of course, for this syntax to work, `lmswitch` needs to be installed as a foreign command.

The new logfile will be opened for write, rather than append, so it is possible to "switch" to the same filename in order to be able to view the old log file.

6.10 lmswitchr

The `lmswitchr` utility switches the report writer (REPORTLOG) log file. It will also start a new REPORTLOG file if one does not already exist.

Usage is:

```
lmswitchr [-c license_file] feature new-file
lmswitchr [-c license_file] vendor new-file [v5.0+ only]
```

where:	**is the:**
-c	license file path
feature	any feature this daemon supports
new-file	new file path

Note—lmswitchr does not work with FLEX*lm* v3.0 vendor daemons. Ask your vendor for a later version of their vendor daemon.

6.11 lmver

The `lmver` utility reports the FLEX*lm* version of a library or binary file.

Usage is:

`lmver` *`filename`*

where:	**is the:**
filename	name of the executable of the product.

For example if you have an application called "spell" type:

```
% lmver spell
```

Alternatively, on Unix systems, you can use the following commands to get the FLEX*lm* version of a binary:

```
strings file | grep Copy
```

6.12 License Administration Tools for Windows

6.12.1 License Administration Tools—LMTOOLS for Windows

For the Windows Platforms, an LMTOOLS.EXE Windows program is provided. It has the same functionality as listed in the previous sections but is graphically-oriented. Simply run the program and choose a button for the functionality required. Refer to the previous sections for information about the options of each feature.

6.12.2 The FLEX*lm* Control Panel

The FLEX*lm* control panel, FLEXLM.CPL, is an applet that installs into the Control Panel of Windows and is used to control the execution of the FLEX*lm* license manager. If the software you are installing includes the Control Panel, it will be installed in your Windows System Directory.

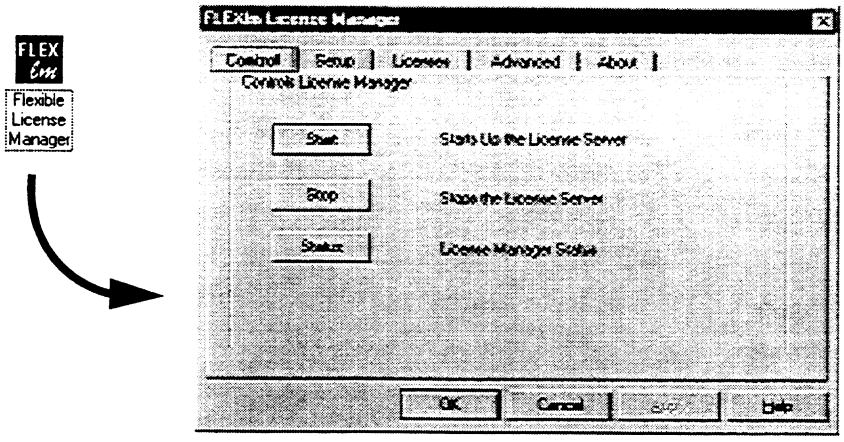
The library LMGR325B.DLL needs to be available for FLEXLM.CPL. In this example it is placed in the same directory, but it could be placed anywhere in the system search path.

6.12.3 Controlling LMGRD with the Windows Control Panel

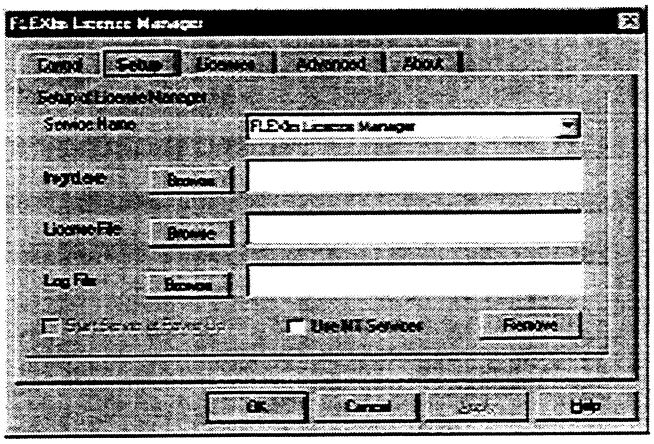
LMGRD.EXE can be run manually or using the Control Panel. FLEX*lm* for Windows comes with a Control Panel applet for controlling LMGRD.EXE without using the DOS prompt.

6.12.4 Using The Control Panel on NT

To start the FLEX*lm* Control Panel, open the Control Panel and double-click on the FLEX*lm* License Manager icon.



From the Control tab you can start, stop, and check the status of your license server. Select the Setup tab to enter information about your license server.



The Service Name of "FLEX*lm* License Manager" is the default (for backwards compatibility). You should change this to a name that your vendor recommends. Complete the form to configure *lmgrd* to serve licenses.

The information you enter is stored in the registry under the service name you created:

`HKEY_LOCAL_MACHINE\SOFTWARE\FLEXlm License Manager\`***`Service-Name`***`\...`

Select the *Control* tab and click the *Start* button to turn on your license server. LMGRD.EXE will be launched as a background application with the license file and debug log file locations passed as parameters.

If you want LMGRD.EXE to start automatically, select the "Use NT Services" box and LMGRD.EXE will be installed as an NT service. You can then use the NT's Services control panel to adjust the start/stop behavior of LMGRD.EXE. Since NT services do not have command line parameters, LMGRD.EXE, when started as a service, locates its service name under "FLEX*lm* License Manager" in the registry and from there recovers the license file and log file locations. Multiple instances of LMGRD.EXE can be run as services provided each occurrence has a different service name.

You can switch back and forth between different instances of LMGRD.EXE by using the *Setup* tab and changing the selection in "Service Name". This is only necessary if you have more than one product licensed with FLEX*lm*.

The remaining tabs in the control panel allow you a subset of control similar to the LMUTIL.EXE program. The Licenses tab provides information about the license file and the Advanced tab allows you to perform diagnostics and check versions.

6.12.5 Using the Control Panel on Windows 95

The behavior of the control panel applet FLEXLM.CPL is almost identical under Windows 95 and NT. FLEXLM.CPL is located in the Windows\System directory. If you are starting LMGRD.EXE manually from the *Control* tab, there is no difference between the two. But, since services are not available on Windows 95, the *Use NT Service* check box is not available. Instead a *Start Server at Power-UP* check box gives you the option to start the server when the system is booted.

On Windows 95, FLEX*lm* makes use of a registry feature that launches programs automatically. The "Microsoft\Windows\CurrentVersion\RunServices" registry is used to launch the program LMGRD95.EXE at power-on. This program scans the "FLEX*lm* License Manager" area of the registry and launches an instance of LMGRD.EXE for each service-name it finds.

Note: If someone switches users (i.e. selects "shutdown" and chooses "close all programs and log on as a different user") on Windows 95, all instances of LMGRD.EXE will be terminated (see Microsoft documentation). This is one of the reasons we do not recommend using Windows 95 as a license server

Appendix A

Hostids for FLEX*lm*-Supported Machines

FLEX*lm* uses different machine identifications for different machine architectures. For example, all Sun Microsystems, Inc. machines have a unique hostid, whereas all DEC machines do not. For this reason, the ethernet address is used on some machine architectures as the "hostid". An ethernet address is a 6-byte quantity, with each byte specified as two hexadecimal digits. Specify all 12 hex digits when using an ethernet address as a hostid. For example, if the ethernet address is 8:0:20:0:5:ac, you would specify "0800200005ac" as the hostid.

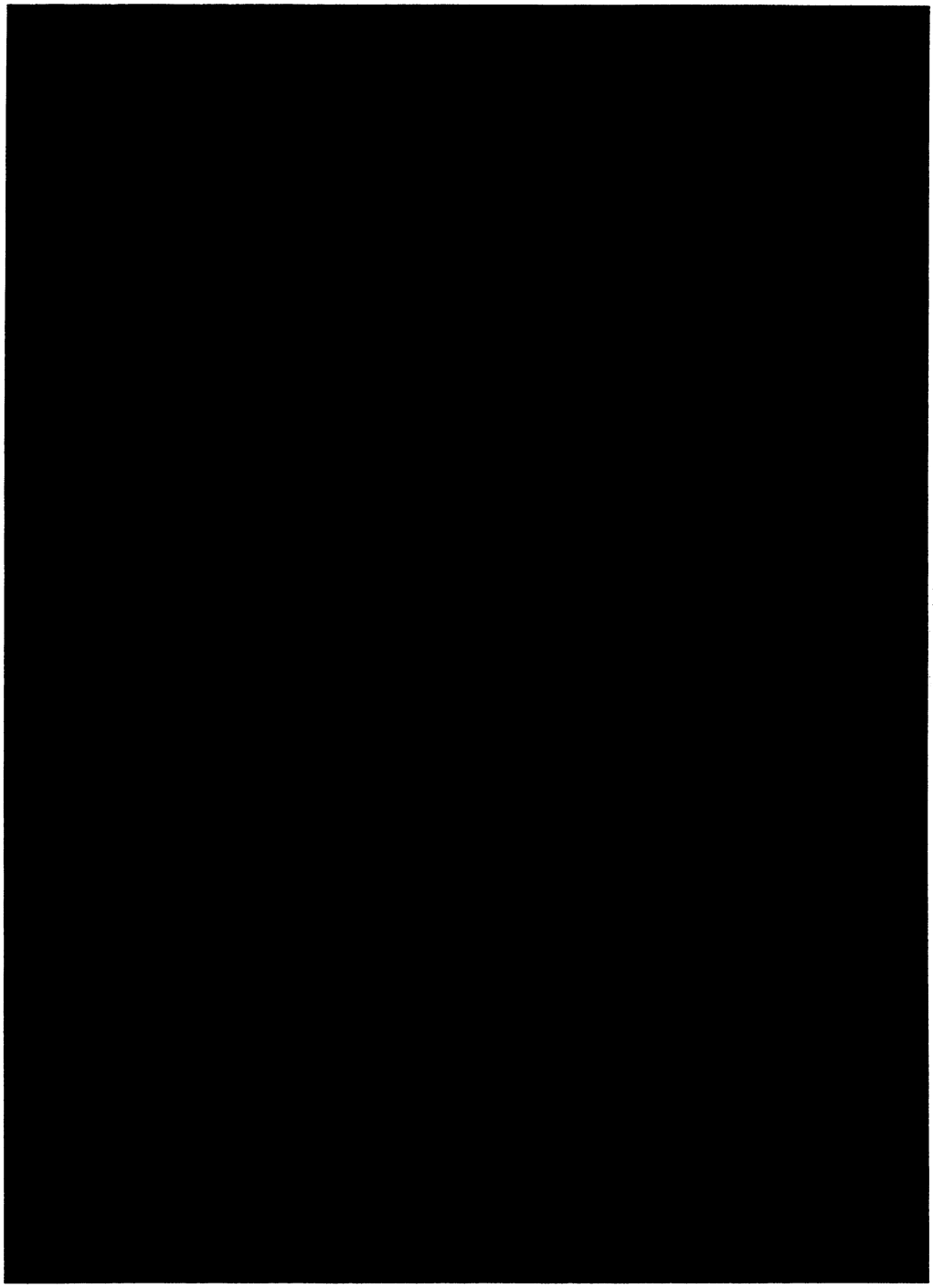
The program `lmhostid` will print the exact hostid that FLEX*lm* expects to use on any given machine. The following table lists alternate methods to obtain the required hostid for each machine architecture

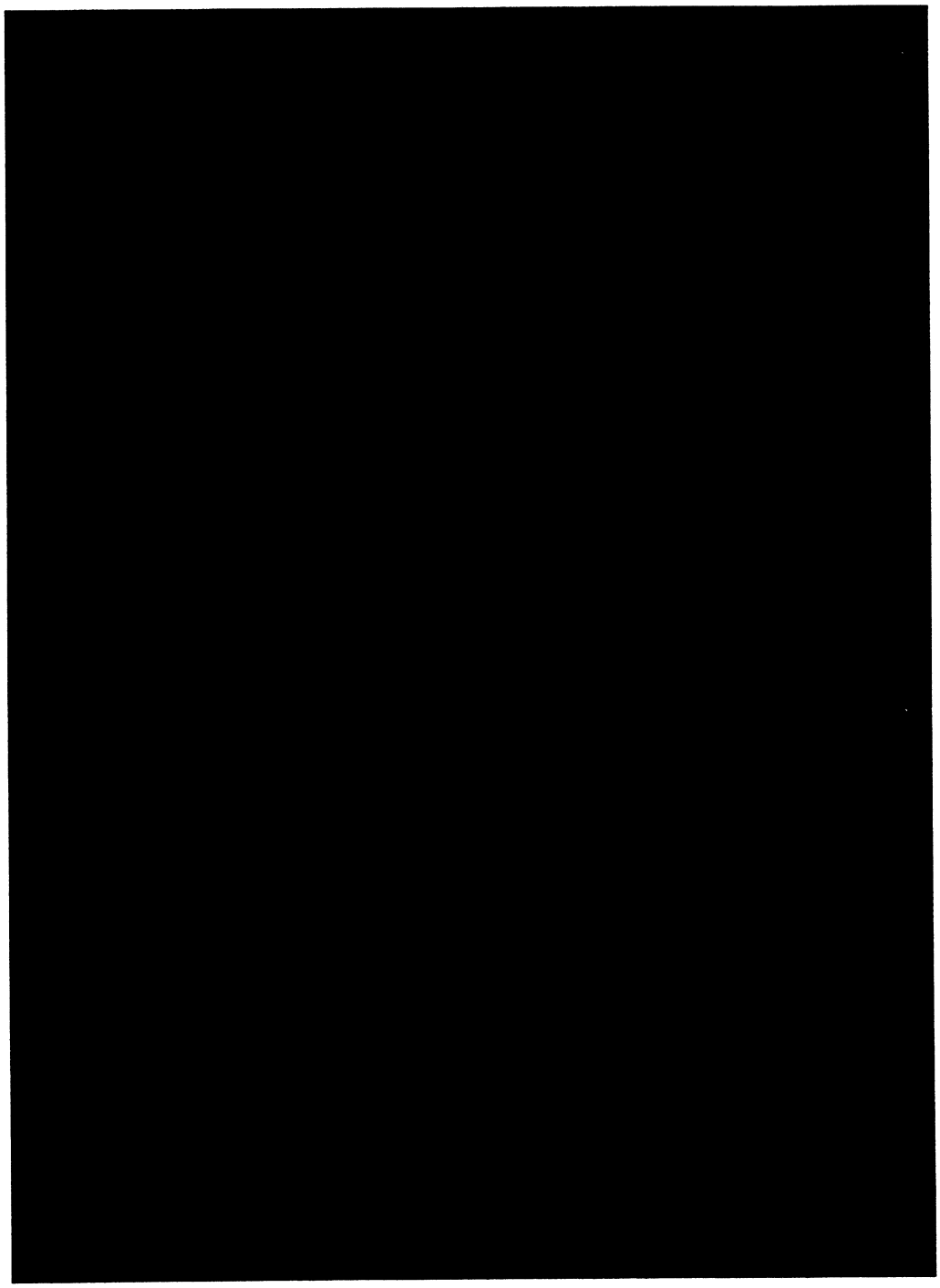
Numeric, 32-bit hostids are normally used in hexadecimal format. On some systems, including HP and SGI, the system command returns the number in decimal format. Since v3.0 of FLEX*lm*, a '#' before the hostid, indicates to FLEX*lm* that this is a decimal number. For example, if the HP uname -i command returns "2005771344", FLEX*lm* will accept "#2005771344". Or it can be converted to hexadecimal. On Unix system, you can convert to hex with the following script:

```
% echo 2005771344 16o p | dc
778DA450
```

A.1 Hostid Table.

Hardware Platform	Hostid	Type this command on the license server:	Example
Apollo	20-bit ID	`lcnode -me`	BE70
DEC Alpha	ethernet address	`netstat -i`	080020005532
Data General	32-bit hostid	`/usr/sbin/systemid`	0C020972





A Hostids for FLEXlm-Supported Machines

Appendix B

Troubleshooting Guide

This appendix documents areas of FLEX*lm* that have given customers difficulty in the past. We hope it helps you debug any problems you might experience at your site.

B.1 General Debugging Hints

The following are tips for debugging:

- When you start the license server (*lmgrd*) be sure that you direct the output into a log file where you can examine it. The log file often contains useful information. You should examine it when you have a problem, and be prepared to answer questions about it when you talk to a support person.
- If the license server appears to have started correctly (which you should be able to determine from the log file), try running `lmstat -a` and `lmdiag` to see if that program has the same problem as your application.
- If your application is FLEX*lm* v4.1 or later (v5 on Windows), you can use the FLEXLM_DIAGNOSTICS environment variable. Set FLEXLM_DIAGNOSTICS to 1 or 2, where 2 gives more information than 1, in particular, the feature name that was denied.

When you talk to a support person, you should be prepared to answer the following questions:

- What kind of machine is your license server running on? What version of the operating system? What machine and operating system is the application running on?
- What version of FLEX*lm* does the program use? Use the `lmver` script, or, on Unix, execute the following command on your *lmgrd*, vendor daemon, and application

  ```
  strings <program> | grep Copy
  ```

  Alternatively, lmgrd -v gives the version, and this works with the vendor daemon also.
- What error or warning messages appear in the log file? Did the server start correctly? Look for a message such as:

  ```
  server xyz started for: feature1 feature2.
  ```
- What is the output from running `lmstat -a`?
- Are you running other products which are also licensed by FLEX*lm*? Are you using a combined license file or separate license files?

- Are you using redundant servers (multiple SERVER lines in your license file)?

B.2 FLEXLM_DIAGNOSTICS

Note—Available only with applications using FLEX*lm* v4.1 or higher for Unix, and v5.0 or higher with Windows. Also, applications may choose not to provide this functionality.

FLEXLM_DIAGNOSTICS is an environment variable that will cause the application to produce diagnostic information when a checkout is denied. The format of the diagnostic information may change over time.

To set FLEXLM_DIAGNOSTICS, on Unix:

```
(csh): % setenv FLEXLM_DIAGNOSTICS 1
(sh): $ FLEXLM_DIAGNOSTICS=1; export FLEXLM_DIAGNOSTICS
```

On Windows 3.1 and 95, add the following line to C:\autoexec.bat:

```
SET FLEXLM_DIAGNOSTICS=1
```

On NT, use the System Control Panel applet to change the global environment, adding FLEXLM_DIAGNOSTICS to 1.

On Unix, the diagnostic output goes to *stderr*.

On Windows, if the application is v5.11 or higher, the output is a file in the current directory called `flex`*nnn*`.log`, where *nnn* is the application's process ID. If the application is v5.0, the output file is called `flex_err.log`.

B.2.1 Level 1 Content

If FLEXLM_DIAGNOSTICS is set to 1, then the standard FLEX*lm* error message will be presented, plus a complete list of license files that the application tried to use. For example:

```
% setenv FLEXLM_DIAGNOSTICS 1
FLEXlm checkout error: Cannot find license file (-1,73:2) No such file
or directory
license file(s):  /usr/myproduct/licenses/testing.dat license.dat
```

B.2.2 Level 2 Content

If FLEXLM_DIAGNOSTICS is set to 2, then, in addition to level 1 output, the checkout arguments are presented. For example:

```
% setenv FLEXLM_DIAGNOSTICS 2
FLEXlm checkout error: No such feature exists (-5,116:2) No such file or
directory
license file(s):  /usr/myproduct/licenses/testing.dat license.dat
lm_checkout("f1", 1.0, 1, 0x0, ..., 0x4000)
```

Note that the error message actually contains 2 separate problems, which both occurred during the checkout: there's no such feature in the license it did find, and it was unable to find the other license file, which is what produces the message "No such file or directory".

Following is a description of the arguments to *lm_checkout()*

`lm_checkout(feature_name, version, nlic, queue_flag, ..., dupgroup_mask)`

where:	**is:**
feature_name	the requested feature
version	requested version. The license file must contain a version >= the requested version.
nlic	number of licenses requested. Usually 1.
queue_flag	If 0, no queueing If 1, queue for license ("blocking" queue) If 2, queue for licenses, but return to application ("non-blocking" queue)
dupgroup_mask	Indicates duplicate grouping, also called license sharing. User, host and display are as shown by lmstat -a. If 0x4000, no license sharing 0x0—site license. All licenses are shared (only 1 licenses at most is ever checked out) 0x1—share if the same user 0x2—share if the same host 0x4—share if the same display 0x8—sharing defined by software vendor These "masks" are usually OR'd together to produce combinations. For example 0x7 means share if the same user and host and display.

B.3 FLEX*lm* Troubleshooting List

B.3.1 Problem Description Format

Each problem is presented in three parts:

Symptom: *A description of the problem.*

Cause: A discussion of what causes the problem described in the "Symptom" section.

Solution: Instructions on how to solve the problem.

You can scan through the list of problems to find any which appear to relate to your concerns. In order to solve your problem, you may have to use all or some of the solutions listed here.

B.3.2 Hostid Problems

Symptom: *When I run the license manager on my machine, it tells me it is the wrong hostid.*

Cause: The vendor daemon checks the hostid listed on the SERVER line in the license file; if it does not match the hostid of the machine it is running on, this message will be printed. Possible causes include 1) you are trying to run the license server on a different machine from the machine the file was made for; 2) the hostid of the machine you are running on changed (for example, the HP ID module was moved, or the CPU board was replaced); 3) the hostid in the license file was modified.

Solution: Verify that the hostid of the machine on which the vendor daemon (or node locked client program) is being run matches the hostid specified in the license file (on the SERVER line for the vendor, or on the FEATURE line for a node locked client). You can run the `lmhostid` program to see what FLEX*lm* thinks the hostid is. You may not modify the hostid in the license file. If the hostid of your server machine changes, you will have to get a new license file from your software vendor.

B.3.3 Connection Problems

Symptom: *The application program (or lmstat) can't connect to the server to check out a license.*

Cause: The FLEX*lm* routines in the application are unable to make a TCP connection to the server and port specified in the license file. Possible reasons for this are: 1) the wrong license file is being referenced by the application program; 2) the server machine specified in the license file is down; 3) the vendor daemon specified in the license file is not running; 4) the hostname in the license file is not recognized by the system; 5) the network between the client machine and the server machine is down; 6) You are mixing FLEX*lm* v1.5 (or earlier) and v2.1 (or later) vendor daemons in one license file, and have run *lmgrd* without the -b command line option; 7) TCP is not running on your machine.

Solution: The lmdiag utility is designed primarily to debug this problem, so first, try lmdiag. Verify that the application is using the proper license file. Verify that specified server machine is up and reachable by executing another command that uses TCP, such as `telnet`, from the client to the server. Verify that the vendor daemon is running (you can use `ps` on the server to look for it). Examine the license log file to see if any problems are reported, particularly messages indicating that the vendor daemon has quit. Run `lmstat -a` from the server machine to verify that the vendor daemon is alive. Run `lmstat -a` from the client machine to

verify the connection from client to vendor daemon across the network. Try using `telnet <hostname> <portnum>` where `hostname` and `portnum` are the same as on the SERVER line in your license file.

B.4 Other Client Problems

Symptom: *When I run my application program (or vendor daemon), I get the error* `bad code` or `inconsistent encryption code`.

Cause: Possible causes for this are 1) the license file was modified (either the hostid on a SERVER line or anything on the FEATURE line was changed); 2) the vendor used the wrong version of his license creation program to generate your license file (or there is a bug in that process).

Solution: You may not modify the license file (except for specific fields, see Chapter 2, "The License File" on page 7). If you need to change something in your license file, you must get a new license file from your software vendor.

Symptom: *When the second user tries to check out a license, the vendor daemon prints an error concerning* `Parameter mismatch` *in the log file and refuses the license.*

Cause: The most likely cause of this problem is that you are simultaneously trying to run two different versions of the application program, and the software vendor has not specifically set up the new version for this kind of compatibility. Check the license server log file for a `comm version mismatch` warning message; this indicates that someone is running an older client than the license server daemon, *lmgrd*.

Solution: Run only the new version of the application (or only the old version).

B.5 Other Server Problems

Symptom: *When I run the vendor daemon on my VMS system, I get the error message* `socket bind: permission denied (13)`.

Cause: The daemon needs to bind the socket address in order to be able to listen for connections from clients. This is done in the system name table, so it requires the SYSNAM privilege.

Solution: Run the daemon in a process with the SYSNAM privilege set.

Symptom: *When I start up lmgrd, it says* `execl failed` *on my vendor daemon.*

Cause: *lmgrd* uses `execl` to start each vendor daemon running. If there is a problem starting the vendor daemon, this message is output to the log file. This error is typically caused by one of the following: 1) there is no executable at the location referred to by the license file (and printed out in the log file); 2) the executable does not have the proper permissions to be run (the file does not have the "x" bit set, or one of the directories in the path is not readable); 3) there was an error building the executable, and it can not be run; 4) the executable is for a different machine architecture.

Solution: Verify that the path to the vendor daemon is absolute (i.e. starts with a slash character, and that it points to the executable program itself, not the containing directory (for FLEX*lm* v1.5). Ensure that the file exists by doing an `ls -l` of the vendor daemon filename(s) specified in the log file. Make sure you do this as the same user that started *lmgrd*. Verify that the file is executable. Note that if you are running as root and using an NFS-mounted filesystem, the relevant protection bits are the "other" bits (*not* the "user" bits), even if the file is owned by root. Do a `whatis` on the file (if your system has that program). `whatis` should tell you the file is an executable for the machine you are trying to run it on. Run the vendor daemon directly from the command line. If the vendor daemon is properly linked, it will tell you that it must be run from *lmgrd*; if it crashes or fails to execute, then it is not properly linked.

Symptom: *The license server keeps reporting "lost lock" errors in the log file and exiting.*

Cause: The lockfile (normally placed in `/usr/tmp` on Unix, `C:\flexlm` on Windows/NT, `SYS:\SYSTEM\FLEXLM` on Netware) is being removed by someone else. There could be another daemon running, or the license administrator (or a script he set up) could have deleted the file.

Solution: Check to see if there is more than one copy of the daemon running. On Unix use a command like `ps -aux | grep vendorname` to search for it. Check for more than one *lmgrd* running as well, since it will restart your vendor daemon when it is killed. If more than one *lmgrd* is running, kill them all (using the `kill` command, not `kill -9` on Unix or the Control Panel Services dialog on Windows/NT), then kill any remaining vendor daemons (on Unix, try a simple kill, if that fails then try `kill -9 the lmgrd` and all vendor daemons) and start one fresh copy of *lmgrd*. On Unix, check to see if there is a shell script running that cleans out `/tmp` (or `/usr/tmp`). If so, try modifying it so that it does not delete zero length files.

Appendix C

Frequently Asked Questions

C.1 License File Questions

C.1.1 I've received FLEXlm license files from 2 different companies. Do I have to combine them?

You don't have to combine license files. Each license file that has any "counted" lines (the "number of licenses" field is >0) requires a server. It's perfectly OK to have any number of separate license files, with different lmgrd server processes supporting each file. Moreover, since lmgrd is a lightweight process, for sites without system administrators, this is often the simplest (and therefore recommended) way to proceed.

C.1.2 When is it recommended to combine license files?

Many system administrators, especially for larger sites, prefer to combine license files to ease administration of FLEX*lm* licenses. It's purely a matter of preference.

C.1.3 Does FLEX*lm* handle dates in the year 2000 and beyond?

Yes. The FLEX*lm* date format uses a 4-digit year. Dates in the 20th century (19xx) can be abbreviated to the last 2 digits of the year (xx), and use of this feature is quite widespread. Dates in the year 2000 and beyond must specify all 4 year digits.

C.2 FLEX*lm* Versions

C.2.1 I have products from several companies at various FLEX*lm* version levels. Do I have to worry about how these versions work together?

If you're not combining license files from different vendors, the simplest thing to do is make sure you use the tools (especially lmgrd) that are shipped by each vendor.

lmgrd will always correctly support older versions of vendor daemons and applications, so it's ALWAYS safe to use the latest version of lmgrd. If you've combined license files from 2 vendors, you MUST use the latest version of lmgrd.

If you've received 2 versions of a product from the same vendor, you MUST use the latest vendor daemon they send you. An older vendor daemon with a newer client will cause communication errors.

Please ignore letters appended to FLEX*lm* versions, i.e., v2.4d. The appended letter indicates a patch, and does NOT indicate any compatibility differences. In particular, some elements of FLEX*lm* didn't require certain patches, so a 2.4 lmgrd will work successfully with a 2.4b vendor daemon.

C.2.2 I've received a new copy of a product from a vendor, and it uses a new version of FLEX*lm*. Is my old license file still valid?

Yes. Older FLEX*lm* license files are always valid with newer versions of FLEX*lm*.

C.2.3 I've received a new license file, and the format is different from the old one I had from the same vendor? Why? Are they compatible?

As of v3.0, FLEX*lm* has an optional new format for license files, which is explained fully in Q3.1. FLEX*lm* products always understand older versions; therefore, the pre-v3.0 files are understood by every FLEX*lm* version. However, your old applications (pre-FLEX*lm* v3.0) will not be able to use the new license file.

C.3 Using FLEX*lm*

C.3.1 Does FLEX*lm* work across the internet?

Yes. A server on the internet will serve licenses to anyone else on the internet. This can be limited with the INTERNET= attribute on the FEATURE line, which limits access to a range of internet addresses. You can also use the INCLUDE and EXCLUDE options in the daemon option file to allow (or deny) access to clients running on a range of internet addresses.

C.3.2 Does FLEX*lm* work with Internet firewalls?

Many firewalls require that port numbers be specified to the firewall. FLEX*lm* v5 lmgrd supports this.

C.3.3 If my client dies, does the server free the license?

Yes, unless the client's whole system crashes. Assuming communications is TCP, the license is automatically freed immediately. If communications are UDP, then the license is freed after the UDP timeout, which is set by each vendor, but defaults to 45 minutes. UDP communications is normally only set by the end-user, so TCP should be assumed. If the whole system crashes, then the license is not freed, and you should use lmremove to free the license.

C.3.4 What happens when the license server dies?

FLEX*lm* applications send periodic heartbeats to the server to discover if it has died. What happens when the server dies is then up to the application. Some will simply continue periodically attempting to re-checkout the license when the server comes back up. Some will attempt to re-checkout a license a few times, and then, presumably with some warning, exit. Some GUI applications will present pop-ups to the user periodically letting them know the server is down and needs to be re-started.

C.3.5 How do you tell if a port is already in use?

99.44% of the time, if it's in use, it's because lmgrd is already running on the port - or was recently killed, and the port isn't freed yet. Assuming this is not the case, then use 'telnet host port'— if it says "can't connect", it's a free port.

C.3.6 Does FLEX*lm* require root permissions?

No. There is no part of FLEX*lm*, lmgrd, vendor daemon or application, that requires root permissions. In fact, it is *strongly recommended* that you do not run the license server (lmgrd) as root, since root processes can introduce security risks. If lmgrd must be started from the root user (for example, in a system boot script), we recommend that you use the *su* command to run lmgrd as a non-privileged user:

```
su username -c "/path/lmgrd -c /path/license.dat -l /path/log"
```

where *username* is a non-privileged user, and *path* is the correct paths to lmgrd, license.dat and debug log file. You will have to ensure that the vendor daemons listed in */path-to-license/license.dat* have execute permissions for *username*. The paths to all the vendor daemons in the license file are listed on each DAEMON line.

C.3.7 Is it ok to run lmgrd as "root" (Unix only)?

It is not prudent to run any command, particularly a daemon, as root on Unix, as it may pose a security risk to the Operating System. Therefore, we recommend that *lmgrd* be run as a non-privileged user (not "root"). If you are starting lmgrd from a boot script, we recommend that you use

```
su username -c "umask 022; lmgrd..."
```

to run *lmgrd* as a non-privileged user.

C.3.8 Does FLEX*lm* licensing impose a heavy load on the network.

No, but partly this depends on the application, and end-user's use. A typical checkout request requires 5 messages and responses between client and server, and each message is < 150 bytes.

When a server is not receiving requests, it requires virtually no CPU time.

When an application, or lmstat, requests the list of current users, this can significantly increase the amount of networking FLEX*lm* uses, depending on the number of current users.

Also, prior to FLEX*lm* v5, use of "port@host" can increase network load, since the license file is down-loaded from the server to the client. port@host should be, if possible, limited to small license files (say < 50 features). In v5, port@host actually *improves* performance.

C.3.9 Does FLEX*lm* work with NFS?

Yes. FLEX*lm* has no direct interaction with NFS. FLEX*lm* uses an NFS-mounted file like any other application.

C.3.10 Does FLEX*lm* work with ATM, ISDN, Token-Ring, etc.

In general, these have no impact on FLEX*lm*. FLEX*lm* requires TCP/IP or SPX (Novell Netware). So long as TCP/IP works, FLEX*lm* will work.

C.3.11 Does FLEX*lm* work with subnets, fully-qualified names, multiple domains, etc.?

Yes, although this behavior has been improved in v3.0.

When a license server and a client are located in different domains, fully-qualified host names have to be used. A "fully-qualified hostname" is of the form: node.domain, where "node" is the local hostname (usually returned by the 'hostname' command or 'uname -n') "domain" is the internet domain name, e.g., "globes.com".

To ensure success with FLEX*lm* across domains, do the following:

1. Make the sure the fully-qualified hostname is the name on the SERVER line of the license file
2. Make sure ALL client nodes, as well as the server node, are able to 'telnet' to that fully-qualified hostname. For example, if the host is locally called "speedy", and the domain name is "corp.com", local systems will be able to logon to speedy via 'telnet speedy'. But very often, 'telnet speedy.corp.com' will fail, locally. Note that this telnet command will always succeed on hosts in other domains (assuming everything is configured correctly), since the network will resolve speedy.corp.com automatically.
3. Finally, there must be an "alias" for speedy so it's also known locally as speedy.corp.com. This alias is added to the /etc/hosts file, or if NIS/Yellow Pages are being used, then it will have to be added to the NIS database. This requirement goes away in version 3.0 of FLEX*lm*.

C.3.12 Does FLEX*lm* work with NIS and DNS?

Yes. However, some sites have broken NIS or DNS, which will cause FLEX*lm* to fail. In v5 of FLEX*lm*, NIS and DNS can be avoided to solve this problem. In particular, sometimes DNS is configured for a server that's not current available (e.g., a dial-up connection from a PC). Again, if DNS is configured, but the server is not available, FLEX*lm* will fail.

In addition, some systems, particularly Sun, SGI, HP, require that applications be linked dynamically to support NIS or DNS. If a vendor links statically, this can cause the application to fail at a site that uses NIS or DNS. In these situations, the vendor will have to relink, or recompile with v5 FLEX*lm* (when it becomes available in Q1 of 96). Vendors are strongly encouraged to use dynamic libraries for libc and networking libraries, since this tends to improve quality in general, as well as making NIS/DNS work.

On PCs, if a checkout seems to take 3 minutes and then fails, this is usually because the system is configured for a dial-up DNS server which is not currently available. The solution here is to turn off DNS.

Finally, hostnames must NOT have periods in the name. These are not legal hostnames, although PCs will allow you to enter them, and they will not work with DNS.

C.3.13 Is the FLEX*lm* "display" the same as an X-Display?

Not by default. The default FLEX*lm* display is what is returned by the ttyname() function call (or the 'tty' command), and is usually something like "/dev/ttyp4". However, the application developer can change this default to the X-Display. A paper is available on this topic to FLEX*lm* developers from GLOBE*trotter* Software.

C.3.14 We're using FLEX*lm* over a wide-area network. What can we do to improve performance?

FLEX*lm* network traffic should be minimized. With the most common uses of FLEX*lm*, traffic is negligible. In particular, checkout, checkin and heartbeats use very little networking traffic. There are two items, however, which can send considerably more data and should be avoided or used sparingly:

- "lmstat -a" should be used sparingly. lmstat -a should not be used more than, say, once every 15 minutes, and should be particularly avoided when there's a lot of features, or concurrent users, and therefore a lot of data to transmit; say, more than 20 concurrent users or features.
- prior to FLEX*lm* v5, the "port@host" mode of the LM_LICENSE_FILE environment variable should be avoided, especially when the license file has a lot of features, or there are a lot of license files included in $LM_LICENSE_FILE. The license file information is sent via the network, and can place a heavy load. Failures due to "port@host" will generate the lm_errno LM_SERVNOREADLIC (-61).

C.4 PC-Related Questions

C.4.1 What PC Platforms are supported?

FLEX*lm* supports Windows 3.1,Windows for Workgroup 3.11, Windows 95, Windows NT 3.5, 3.51 and 4.0 on Intel Mips and Alpha, Netware 3.12 and 4.X, OS/2 3.0.

Networking is required on all 32-bit versions

C.4.2 What is WINSOCKX.DLL

Winsockx.dll is a DLL provided by GLOBE*trotter* Software that is used by 16 bit applications to interface between FLEX*lm* and other networking software provided by networking ISV's. It allows node locked applications to not require networking software. It also interfaces between winsock.dll for TCP/IP, and the Novell DLLs that provide IPX/SPX on 16 bit operating systems.

C.4.3 Why do I need to include NWlink IPX/SPX on NT and Win 95?

This is necessary for either obtaining the Ethernet card address, or to provide connectivity with a Netware License server.

C.4.4 Will the Server run on Windows 95?

Yes, lmgrd runs in it's own window. If available, NT systems are preferred, since it can be run as an NT "service".

C *Frequently Asked Questions*

Appendix D

The Debug Log File

The daemons all generate debug log files in the following format.

```
hh:mm (DAEMON NAME) message
```

where:	is the:
hh:mm	Time that the message was logged.
DAEMON NAME	Either `lmgrd` or the vendor DAEMON name. In the case where a single copy of the daemon cannot handle all of the requested licenses, an optional "_" followed by a number indicates that this message comes from a forked daemon.
message	The text of the message.

The debug log files can be used to:

- Diagnose configuration problems.
- Diagnose daemon software errors.

Note—The debug log file should not be used for usage reporting.

D.1 Informational Messages

Connected to node

This daemon is connected to its peer on node "node."

CONNECTED, master is name

The license daemons log this message when a quorum is up and everyone has selected a master.

DENIED: N feature to user

"user" was denied access to "N" licenses of "feature".

EXITING DUE TO SIGNAL nnn

EXITING with code nnn

All daemons list the reason that the daemon has exited.

EXPIRED: feature

"feature" has passed its expiration date.

IN: feature by user (N licenses)

"user" has checked back in "N" licenses of "feature".

License Manager server started

The license daemon was started.

Lost connection to host

A daemon can no longer communicate with its peer on node "host", which can cause the clients to have to reconnect, or cause the number of daemons to go below the minimum number, in which case clients may start exiting. If the license daemons lose the connection to the master, they will kill all the vendor daemons; vendor daemons will shut themselves down.

Lost quorum

The daemon lost quorum, so will process only connection requests from other deamons.

MULTIPLE xxx servers running.

Please kill, and restart license daemon

The license daemon has detected that multiple licenses for vendor daemon "xxx" are running. The user should kill all "xxx" daemon processes and re-start the license daemon.

OUT: feature by user (N licenses)

"user" has checked out "N" licenses of "feature".

RESERVE feature for HOST name

RESERVE feature for USER name

A license of "feature" is reserved for either user "name" or host "name".

REStarted xxx (internet port nnn)

Vendor daemon "xxx" was restarted at internet port "nnn".

Retrying socket bind (address in use)

The license servers try to bind their sockets for approximately 6 minutes if they detect "address in use" errors.

Selected (EXISTING) master node.

This license daemon has selected an existing master (node) as the master.

SERVER shutdown requested.

A daemon was requested to shut down via a user-generated kill command.

[NEW] Server started for: feature-list

A new server was started for the features listed.

Shutting down xxx

The license daemon is shutting down the vendor daemon xxx.

SIGCHLD received. Killing child servers.

A vendor daemon logs this message when a shutdown was requested by the license daemon.

Started name

The license daemon logs this message whenever it starts a new vendor daemon.

Trying connection to node

The daemon is attempting a connection to "node".

D.2 Configuration Problem Messages

***hostname*: Not a valid server host, exiting**

This daemon was run on an invalid hostname.

***hostname*: Wrong hostid, exiting**

The hostid is wrong for "hostname."

BAD CODE for *feature-name*

The specified feature name has a bad license key. It was probably typed in wrong, or modified by the end-user.

CANNOT OPEN options file *"file"*

The options file specified in the license file could not be opened.

license daemon: lost all connections

This message is logged when all the connections to a server are lost, which often indicates a network problem.

lost lock, exiting

Error closing lock file

Unable to re-open lock file

The vendor daemon has a problem with its lock file, usually because of an attempt to run more than one copy of the daemon on a single node. Locate the other daemon that is running via a `ps` command, and kill it with `kill -9`.

NO DAEMON line for *daemon*

The license file does not contain a "DAEMON" line for "*daemon.*"

No "license" service found

The TCP "license" service did not exist in `/etc/services`.

No features to serve!

A vendor daemon found no features to serve. This could be caused by a corrupted or incorrectly entered license file.

UNSUPPORTED FEATURE request: feature by user

The "user" has requested a feature that this vendor daemon does not support. This can happen for a number of reasons: the license file is bad, the feature has expired, or the daemon is accessing the wrong license file.

Unknown host: hostname

The hostname specified on a "SERVER" line in the license file does not exist in the network database.

NO DAEMON lines, exiting

The license daemon logs this message if there are no DAEMON lines in the license file. Since there are no vendor daemons to start, there is nothing to do.

NO DAEMON line for name

A vendor daemon logs this error if it cannot find its own DAEMON name in the license file.

D.3 Daemon Software Error Messages

No internet port number specified

A vendor daemon was started without an internet port.

read: error message

An error in a "read" system call was detected.

select: message

An error in a select system call was detected. This is usually a sign of a system networking failure.

Server exiting

The server is exiting. This is normally due to an error.

Appendix E

FLEX*lm* Versions

V1.0—1988

First FLEX*lm* Release, containing all the basic FLEX*lm* features.

V1.5—February 1990

First widely-used version including DEMO.

V2.1—March 1991

- Improved TIMEOUT support
- Improved ethernet hostid support

V2.21—November 1991

- Added support for many platforms, and some platform- specific improvements, such as hostid.
- Hostid ANY added

V2.26—March 1992 (Used only by Sun)

- Added license lingering

V2.4—December 1992

- Added "use-all-feature-lines" capability for incremental license distribution.
- Enhanced vendor customization routines.
- Enhanced End-User Options file.
- Added new hostid types: USER, HOSTNAME and DISPLAY
- Added "port@host" to locate license file —down-loads license file from server.

V2.61—March 1993 (Used only by Sun)

- Added INCREMENT and UPGRADE lines to license file

V3.0—May 1994

- INCREMENT and UPGRADE behavior changed and improved
- Added UDP protocol support.
- Added 'uname -i' hostid for hp.
- Added 'multiple jobs,' for enhanced support of LM_LICENSE_FILE as a license file list.

- New, optional license file format, with "name=value" syntax for optional new features, including: asset_info, ISSUER, and NOTICE, '\' license file continuation character, 2048 character limit per feature

V4.0—December 1994

- Removed use of floating point, for enhanced reliability
- FEATURE line additions: ck (for use with lmcksum), OVERDRAFT, DUP_GROUP, INTERNET hostid
- PACKAGE line
- License Finder
- lmdiag and FLEXLM_DIAGNOSTICS for end-user diagnostics

V4.1—May 1995

- Performance improvements and new platform support.

V4.1—Patch Release 6, October 1995

- PC patch release for Windows 95 with various performance improvements.

V5.0—March 1996

- Improved port@host behavior—client application doesn't read license file.
- Automatic port@host via USE_SERVER line in license file.
- Hostid lists—lock a feature to several hostids.
- New FEATURE attributes: SN (serial number), USER_BASED, HOST_BASED, MINIMUM, SUPERSEDE, ISSUED (issued date), CAPACITY (charging based on system capacity).
- Optional avoidance of NIS and DNS via IP address instead of hostname.
- Improved REPORTLOG format
- Server, upon startup, notifies of licenses that will expire within 2 weeks.
- Improved end-user options file functionality.
- New FEATURE attributes: SUPERSEDE, ISSUED, USER_BASED, HOST_BASED, SN.

V5.11 - February, 1997

- SUPERSEDE lists, PLATFORMS= license attribute,
- new end-user options: MAX, TIMEOUTALL, CPU_USAGE_INTERVAL, CPU_USAGE_DELTA
- Windows control panel added.
- Windows license generator GENLIC added.

Index

MARC
Analysis
Research
Corporation

Document Title: **MARC and Mentat Release Notes**
Part Number: RL-3871-31
Revision Date: September, 1997

CALIFORNIA - Palo Alto
MARC Corporate Headquarters
260 Sheridan Avenue, Suite 309
Palo Alto, CA 94306, USA
Tel: 1 650 329 6800
Fax: 1 650 323 5892
Email: support@marc.com

CALIFORNIA - San Diego
MARC Analysis Research Corp.
4330 LaJolla Village Drive, Suite 320
San Diego, CA 92122
Tel: 1 619 658 9588
Fax: 1 619 587 8710

CONNECTICUT - E. Berlin
MARC Analysis Research Corp.
1224 Mill Street
E. Berlin, CT 06023
Tel: 1 860 828-2055
Fax: 1 860 828 2031

MICHIGAN - Ann Arbor
MARC Analysis Research Corp.
24 Frank Lloyd Wright Drive
Ann Arbor, MI 48106-0523
Tel: 1 313 998 0540
Fax: 1 313 998 0542

CZECH REPUBLIC
MARC Overseas, Inc.
Podolska 50
147 00 Praha 4
Czech Republic
Tel: 011 420 (2) 61 21 41 23
Mobil: 011 420 (602) 21 23 26
Fax: 011 420 (2) 61 21 41 23

FRANCE
MARC France S.A.
30, Rue de Rocroy
94100 Saint-Maur Des Fossés (near Paris)
France
Tel: 011 33 (1) 41 81 1 0 94
Fax: 011 33 (1) 41 81 1 0 95

GERMANY - Munich
MARC Software Deutschland GmbH
Ismaninger Str. 9
85609 Aschheim (bie Munich)
Germany
Tel: 011 49 (89) 904 50 33
Fax: 011 49 (89) 903 06 76
Email: system@marc.co.de

GERMANY - Hannover
MARC Software Deutschland Gmbh
Alte Dohrener Str. 66
30173 Hannover, Germany
Tel: 011 49 (511) 980 5182
Fax: 011 49 (511) 980 5187

ITALY
Espri-MARC s.r.l.
16129 Viale Brigata Bisagno 2/10
16129 Genova
Italy
Tel: 011 39 (10) 585 949
Fax: 011 39 (10) 585 949

THE NETHERLANDS
MARC Analysis Research Corporation
Dublinstraat 32
2713 HS Zoetermeer
The Netherlands
Tel: 011 31 (79) 3510 411
Fax: 011 31 (79) 3517 560
Email: support@marc.nl

UNITED KINGDOM
MARC UK Ltd.
35, Shenley Pavilions
Chalkdell Drive
Shenley Wood
Milton Keynes, MK56LB
Tel: 011 44 (1908) 506 505
Fax: 011 44 (1908) 506 522

CHINA
MARC Overseas, Inc.
Bright Chang An Building, Suite 321
7# Jian Guo Men Nei Street
Beijing, China 100 005
Tel: 011 86 (10) 6510 2056
011 86 (10) 6510 2057
011 86 (10) 6510 2058
Fax: 011 86 (10) 6510 2053

JAPAN - Nagoya
Nippon MARC Co., Ltd.
Nagoya Nishiki Daiichi Seimei Bldg.
2F, 6-5 Nishiki 1-chome
Naka-ku, Nagoya, Aichi 460
Tel: 011 81 (52) 202 5661
Fax: 011 81 (52) 202-5665

JAPAN - Osaka
Nippon MARC Co., Ltd.
4F 2nd Kimi Building
2-11 Toyotucho Suita-shi
Osaka 564
Tel: 011 81 (6) 385 1101
Fax: 011 81 (6) 385-4343

JAPAN - Tokyo
(Asian Operations Headquarters)
Nippon MARC Co., Ltd.
Shinjuku Daiichi Seimei Building
P.O. Box 5056
2-7-1 Nishi-Shinjuku, Shinjuku-ku
Tokyo 163
Tel: 011 81 (3) 3345 0181
Fax: 011 81 (3) 3345 1529
Email: system@marc.co.jp

KOREA
Dong Kyung Bldg. 7FL
824-19 Yuksam-Dong
Kangnam-Ku
Seoul, Korea
Tel: 011 82 (2) 561 7543
Fax: 011 82 (2) 561 7767