

MULTI[®] 2000 Installation and Licensing Guide



Copyright © 1983-1999 by Green Hills Software, Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission from Green Hills Software, Inc.

DISCLAIMER

GREEN HILLS SOFTWARE, INC. MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. Further, Green Hills Software, Inc. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Green Hills Software, Inc. to notify any person of such revision or changes.

Green Hills Software and the Green Hills logo are trademarks, and MULTI is a registered trademark, of Green Hills Software, Inc.

System V is a trademark of AT&T.

Sun is a trademark of Sun Microsystems, Inc.

UNIX and Open Look are registered trademarks of UNIX System Laboratories.

ColdFire is a registered trademark of Motorola, Inc.

DEC, VAX, and VMS are trademarks of Digital Equipment Corporation.

4.2BSD is a trademark of the Board of Regents of the University of California at Berkeley.

X and X Window System are trademarks of the Massachusetts Institute of Technology.

Motif is a trademark of Open Software Foundation, Inc.

Microsoft is a registered trademark, and Windows, Windows 95, and Windows NT are trademarks of Microsoft Corporation.

All other trademarks or registered trademarks are property of their respective companies.

The license management portion of this product is based on:

Élan License Manager

© 1989-1997 Élan Computer Group, Inc.

All rights reserved.

PubID: M45U-C1099-89NG

Printed: October 26, 1999 2:43 pm

CONTENTS

	Preface	P-1
	Who Should Read This Guide?	P-2
	What This Guide Contains	P-2
	Conventions	P-2
1	Installation Guide	1
	Important	2
	System requirements	2
	Preparation	2
	How to install your Green Hills products	3
	How to set up the user environment	6
	Creating a new server code	7
	Notes for system administrators	8
2	Requesting and Installing Licenses	9
	Types of Licenses	10
	Requesting and Installing a License	10
	Special Considerations for License Requests	11
	The Floating License Model	13
	Using the Command Line	14
	Getting Help	14
	Retrieving Debug Output	15
	Debugging the Client	15
	Debugging the License Server	15
3	Administrator Commands	17
A	Resource Files	A-1
	Restricting the Domain	A-2
	Changing Default Connection Settings	A-3

CONTENTS

	Listing Redundant Servers	A-3
	Reserving, Excluding, and Holding Licenses	A-4
	Examples	A-5
B	Troubleshooting	7
	What to do if the Elan license server is not working	8
	What to do if you experience delays	15
	Other causes of problems	16
	Index	I-1

Preface

Welcome to the Green Hills License Manager: your self-maintaining license-management system. The software you have just purchased uses the Élan License Manager to administer and control its runtime licenses. You will only need to consult this guide if you are attempting to alter the default configuration of the license manager as established on initial product installation.

Who Should Read This Guide?

This guide is intended for UNIX system administrators responsible for maintaining installations of software applications using the Green Hills License Manager. It assumes that you are familiar with the operating system and standard system administration tasks.

What This Guide Contains

This guide contains three chapters and two appendices.

- Installation Guide contains instructions for installing Green Hills products on a UNIX platform.
- *Requesting and Installing Licenses* describes the License Manager and gives step by step instructions for establishing it.
- *Administrator Commands*, describes the administrative commands.
- Appendix A, *Resource Files*, describes the resource file formats used by the license server.
- Appendix B, *Troubleshooting*, offers suggestions for getting around common problems.

Conventions

This manual uses the following conventions.

Fonts

- Command names (within a paragraph or description) and dialog box names and items are in **boldface**.

elmadmin -c

or

the **Key** field on the **Create Key File** dialog box.

- Command-line examples and information displayed on the system are in courier. Also, arguments for which you supply values are in *italic*.

`client -e keydir -f name`

- Optional elements are in square brackets ([]). Alternative optional elements are separated by a vertical bar (|).

`elmadmin [-b cid -c -C file -d -e keydir -h -i -j n -k -l -p -q -r n -s -v n
-V C -x -z] [features | alias | feature=alias,...]`

Keys

- Key names are spelled as they appear on the keyboard.

CTRL, ALT, ENTER

- Key combinations are represented with a plus sign (+). This sign indicates that you should press the first key in the sequence and hold it while you press the remaining keys.

CTRL+C, CTRL+SHIFT

Installation Guide

This chapter contains:

- Important
- System Requirements
- Preparation
- How to install your Green Hills Products
- How to set up user environment
- Creating a new server code
- Notes for system administrators

This chapter describes how to install Green Hills products on a UNIX platform.

Important

- If you are upgrading from version 1.8.9 to 2.0, **DO NOT INSTALL YOUR 2.0 RELEASE IN THE SAME LOCATION AS YOUR 1.8.9 RELEASE.**
- To use Green Hills products, the Green Hills License Manager must be installed on your network:
 - If you already have the license manager running on your network and need only to install the software distribution, skip Steps 3 and 4.
 - If you have not yet installed the license manager, you need to complete Steps 3 and 4, which include contacting Green Hills Software to obtain a license key (1 business day turn-around for US and Canada).

System requirements

- A Solaris 2.5 or higher SPARC-based workstation with at least 32MB of physical memory, a graphical display running the X Windowing System, and an Ethernet interface with a TCP/IP stack.
- Green Hills recommends 85 MB of free disk space. You will need more disk space for any source code you write.
- A graphical web browser is required to view online help. Green Hills recommends using a browser that supports Java 1.1.

Preparation

Before installing your Green Hills products, please review the following checklist:

- Select a **userid**. Superuser privilege is not required for either the Green Hills installation process or the Green Hills license manager (**ghs_elmd**).

However, superuser privilege is needed to configure your system to autostart the license manager on reboot.

- Choose a path on your file system for the installation. The examples in this guide use `/usr/green`.
- If you are using VxWorks or Chorus products, see the *Development Guide* for your target for complete installation instructions.

Note

If the **userid** you selected is **root**, then the computer you perform the installation on should either physically mount or network mount, with root access, the installation directory. If you have purchased one or more floating licenses, then the installation computer should also be the one which will run the **ghs_elmd**. If the computer you have chosen cannot meet both of these criteria, contact Green Hills Technical Support.

How to install your Green Hills products

Task 1: Create the installation directory

1. Login to the computer you chose for the installation.
2. Become the **userid** you selected.
3. Create a single directory to hold the Green Hills product files. For example, enter:

```
mkdir /usr/green
```

4. Before you run the installation script, you must have read, write, and execute permissions for the installation directory. For example, enter:

```
chmod 755 /usr/green
```

After installation, write permission is no longer needed. If you decide to change the installation directory name, you must first ensure that the directory is writable, then run the installation script again. The installation script takes the full path name of the directory and inserts it in some of the files. It is safe to run the installation script more than once.

Task 2: Mount the Green Hills CD and extract the files

1. Insert the Green Hills distribution CD.
2. Look for the single `tar` file located on the CD.

3. Set the current directory to the installation directory that you created in Task 1: Create the installation directory. For example, enter:

```
cd /usr/green
```

4. Extract the contents of the `tar` file into the installation directory. For example:

```
tar xvf /cdrom/cdrom0/tarfile.tar
```

When the `tar` command finishes, the Green Hills software distribution has been copied to the installation directory.

Task 3: Run the Green Hills installation script

If you already have the license manager running on your network and need only to install the software distribution, skip Tasks 3 and 4.

To run the Green Hills installation script, enter:

```
/ghs_install.sh
```

The Green Hills installation script prompts you through a series of dialog boxes, some of which give you the option to edit init scripts and the option to request a license key.

Editing init scripts

The `init` (initialization) scripts are highly configurable and may be opened up in the text editor; this allows you to customize your licensing and start-up options.

If you want the license manager to be run automatically on system startup, the installer will attempt to configure your `init` files for you.

Requesting a license key

The installer creates a template to be sent to Green Hills for requesting a license key. The installer can automatically send the template for you or you can e-mail or fax the template to Green Hills:

- e-mail to: `license@ghs.com`
- fax to: (805) 965-6343 attn: License Administrator

The Green Hills License Manager is now installed.

Task 4: Install the license key

After receiving your request for a license key, Green Hills will e-mail back to you within one business day (US and Canada) a UNIX script that contains and installs your license key.

1. Save the script attached to the e-mail to your installation directory.
2. Run the script:

```
./scriptname.sh
```

Task 5: Run the installation script

Once you have successfully installed the License Manager, including the license key, you can run the installation script, **install.sh**.

1. To run the installation script, enter:

```
/install.sh
```

2. When prompted to specify the path name that Green Hills tools will use to run each other, confirm the installation directory name. In most cases, you can simply use the default that is displayed. However, if your network configuration requires you to specify a different path name, enter the path name after the following message:

```
Enter an absolute pathname for this directory or press return  
to use /usr/green: <return>
```

If you enter a directory name that does not exist, differs from the current directory, or is not an absolute path, the installation script displays an error message and re-prompts you for another directory name.

How the installation script tests the installed products

The installation script tests the products in the installation directory, displaying the following:

```
Perform delivery tests. Each test should print 'Test passed'.
```

The installation script also runs a test script for each product in the distribution. For example:

```
Delivery test: ccppc.tst
Testing ANSI C ppc601      Test passed
Testing K+R C ppc601      Test passed
Testing ANSI C ppc602      Test passed
Testing K+R C ppc602      Test passed
Testing ANSI C ppc603      Test passed
```

```
Testing K+R C ppc603           Test passed
Testing ANSI C ppc604         Test passed
Testing K+R C ppc604         Test passed
Testing ANSI C ppc500         Test passed
Testing K+R C ppc500         Test passed
Testing ANSI C ppc403         Test passed
Testing K+R C ppc403         Test passed

Delivery test: cxppc.tst
Testing C++ -cpu=ppc601       Test passed
Testing C++ -cpu=ppc603       Test passed
Your installation has successfully completed, as indicated:
Installation completed.
```

How to set up the user environment

The following instructions assume you are running **cs**h (the C shell) or a compatible shell. See your system administrator for appropriate commands if you are running a different shell.

Task 1: Set the product executable search path (optional)

1. To run the Green Hills compiler, assembler, linker or MULTI without specifying a full path name, you must first add the installation directory to your executable search path. For example:

```
set path=($path /usr/green)
```

2. To confirm that the newly installed path is used, enter:

```
rehash
```

Task 2: Set environment variables

1. To successfully use MULTI, you must set the DISPLAY environment variable:

```
setenv DISPLAY hostname:0
```

where *hostname* is the host name of the system on which you want the MULTI windows to be displayed. You can determine your system name with the **hostname** command.

2. If you are running MULTI on a SunOS or Solaris host, set the LD_LIBRARY_PATH and OPENWINHOME environment variables. For example:

```
setenv LD_LIBRARY_PATH /usr/openwin/lib:$LD_LIBRARY_PATH
```

```
setenv OPENWINHOME /usr/openwin
```

3. If you are in a networked environment, and you want to display the MULTI windows on a workstation other than the system on which MULTI is running, you may have to run the **xhost** server access control program. **xhost** must be run on the system on which you want your X Windows to appear. For example:

```
xhost +
```

Creating a new server code

Sometimes, you may need a new license key. For example:

- The license manager needs to be run on a different machine.
- A new software component is installed, such as a debug server, or a new compiler.
- Additional licenses are purchased.

In these cases, you will have to generate a new server code, get a new license key from Green Hills, and install the license key.

Task 1: Generate a server code

The server code uniquely identifies the host CPU. To generate the server code, run **elmadmin -c**. For example:

```
% elmadmin -c
```

Task 2: Get a license key from Green Hills

Once you have your server code, e-mail or fax it to Green Hills.

- e-mail to: license@ghs.com
- fax to: (805) 965-6343 attn: License Administrator

Green Hills will respond to your request for a license within one business day (US and Canada).

Task 3: Install your license key

After receiving your request for a new license key, Green Hills will e-mail back to you within one business day (US and Canada) a UNIX script that contains and installs your license key.

1. Save the script attached to the e-mail to your installation directory.
2. Run the script:

```
./scriptname.sh
```

Notes for system administrators

- All the files and directories in the distribution are contained within a single directory, such as `/usr/green`.
- Write permission to `/usr/green` is not required, except during installation.
- Separate distributions should not be installed into the same directory, unless you are directed to do so by Green Hills.
- Driver scripts, such as **ccppc**, contain an absolute path to the installation directory, placed by **install.sh**. In a networked environment, this path needs to be accessible by all computers.
- If you will be performing the installation as **root**, make sure the installation computer mounts the installation director with superuser privilege. By default, NFS mounts remap **root** to **nobody**, which can cause unexpected behavior.
- You can start the License Manager with any of the following methods, in any combination:
 - *manual starting*. Enter: `/usr/green/Sghs_elmd start`.
 - *rc file starting*. Add `/usr/green/lm` to your system startup scripts. Green Hills recommends that you run the **ghs_elmd** as a non-**root** user, such as `bin`.

The license manager is usually configured to bind to TCP port 8171. If this causes a conflict, please contact Green Hills Technical Support.

- Do not run **SunOS** product binaries on a **Solaris 2.x** system. Green Hills does not support this configuration.

NOTE: On some Solaris 2.x systems, the system's tools needed by the Green Hills development environment are not part of the default OS installation. These tools include the system assembler and linker. Make sure the additional subsets containing these tools are installed on the system before making the Green Hills environment available.

Requesting and Installing Licenses

This chapter contains:

- Types of licenses
- Benefits to you
- Requesting and installing a license
- Special considerations for license requests
- The floating license model
- Using the command line
- Getting help
- Retrieving debug output
- Debugging the client
- Debugging the license server

The Green Hills License Manager is a software package that controls single or multi-user access to software applications in standalone and networked environments. To use our products, you will need to acquire licenses from us. This chapter describes the process for acquiring those licenses, as well as the different types of licenses that are available.

Types of Licenses

- **Floating Licenses:** In a networked environment, floating licenses allow you to centrally administer license usage. Licenses are stored on one or more license servers and can be checked out by users from anywhere on the network. This allows you to add new machines to your network or to have users move from machine to machine without worrying about where the licenses were installed. The primary benefit to floating licenses is that they give you maximum flexibility for administering your licenses. The primary drawback is that it requires TCP/IP support to be installed and configured on all machines.
- **Node-locked Licenses:** In an environment without a network or with only intermittent connectivity, node-locked licenses can be used to ensure that there is no dependence on a network or on other computers. The primary benefit to node-locked licenses is that they are very easy to configure and install. The primary drawback is that the licenses are limited to work only on a particular machine.
- **Dongle, or Hardware Key Licenses:** These licenses are actually a special case of node-locked licenses, but justify special discussion. A dongle is a hardware key that licenses can be locked to. Use of this device allows you to move your licenses from machine to machine just by plugging the dongle into the parallel port on the machine where the tools are to be used. This is useful for users who need to move one set of licenses between several unconnected machines.

Requesting and Installing a License

To request and install a license, follow these three steps:

1. Generate a server code.

Note: This step will be performed automatically when you install MULTI. If you wish to perform this step after you install the product, you can run **xelmadmin** or **elmadmin -c**.

In general, this step should be performed on the machine where you would like to install the license(s). That means that this should be done exactly once for floating licenses on the machine that will act as your license server. For node-locked licenses, it should be done on each workstation that will have a node-locked license.

2. Get a license key from Green Hills.

Once you have your server code, email it to **license@ghs.com** for a license. Alternately, you can use the other methods of contacting Green Hills that are listed on the request.

3. Install your license key.

Green Hills will provide you with instructions for installing your license.

Special Considerations for License Requests

Our license manager is intended to be transparent to the user, but it still requires thoughtful planning. If you want to introduce our tools into a complex development environment, you need to consider the way that your tools are actually used. Each of the license types that we offer have certain advantages and disadvantages, and you want to be careful to choose license types that most closely match the way that you will be using our tools.

To avoid potential problems, please indicate on your license request if any of the following are true:

- You use DHCP or dynamic IP addresses.
- You use PCMCIA or Hot-swappable networking cards.
- You expect the hostname of your machine to change soon or frequently.
- You boot from a removable hard drive.
- You operate from a docking station.
- You expect to have a complex environment utilizing different license types.
- You want to allocate your licenses to more than one license server.

Floating Licenses

If you are using floating licenses, the license server machine should be an easily accessible network host which is up and running as much of the time as possible. In your networking environment, there is probably already one or more server machines that are obvious choices for the license server.

The architecture of floating licenses (described in the next section) requires that TCP/IP is installed and configured correctly. We recommend that you not use floating licenses unless you require the flexibility that they provide. Note that this adds another layer of complexity to your licensing, potentially leading to problems.

Each license transaction requires the client to resolve the IP address of the server from the hostname. Because license transactions are frequent, we recommend that your client machines be configured with a “hosts” file or permanent DNS server.

The floating license directory must be a local directory. For security reasons, the license server will not load licenses from a network directory.

All floating licenses can only be installed on the intended license server. The licenses will not install on other machines. In particular, you do not need to install floating licenses on the client machines.

Dongle Licenses

Green Hills supports the use of a hardware license key, known as a dongle. If you wish to use a single Green Hills license on separate, non-network-attached machines, you can have a node-locked license generated for a dongle, then take your key with you to the machine of your choice. You’ll need to contact your Green Hills Sales representative if you wish to utilize our dongle support.

If you are generating a license request for use with a dongle, be certain that the dongle is attached when the request is generated. Otherwise, the license will be locked to your machine and not the dongle. That means that the licenses will not work on other machines, even if the dongle is attached to them. Dongle license requests should always indicate a server code like “**4194303/...**”

Since dongle licenses are a special case of node-locked licenses, see the section on node-locked licenses.

Node-locked Licenses

Node-locked licenses are generally the easiest to use. They usually operate more quickly because the client doesn’t need to interact with the network for licensing.

Be sure that you specify different directories for your node-locked and floating licenses, or you will get warning messages about not being able to access the license file when you use the tools.

Node-locked licenses need to be installed on the machine that the licenses were created for. Node-locked licenses cannot be installed on other machines. In particular, you do not want to install node-locked licenses on a license server.

The Floating License Model

The Green Hills License Manager uses the client/server model for floating licenses. The license server runs on one or more hosts and the application runs on one or more clients. The license server and client may be the same machine, or they may be different machines (as in a network environment). The server and client do not require access to the same file system.

The license server maintains a central database of licenses and license activity. It verifies clients, grants licenses to valid client applications, and collects the licenses when the applications terminate. The license server runs as a system process, invoked at system start-up.

Normally, the client will issue a network broadcast to find a license server. Note that network broadcasts are by definition unreliable, and are subject to certain limitations. The most notable limitation is that broadcasts are not forwarded through routers. They are also commonly lost on networks with heavy traffic.

If your client and server machines are separated by a router, or you find that the clients do not reliably find the server, you will need to tell the client how to find the server. To do this, you can set the environment variable **GHS_LMHOST**. The value of this variable explicitly indicates the IP hostname(s) or address(es) of the license server(s). If you provide multiple server names, you can separate the names with commas (,), colons (:), or semi-colons (;).

Host names may be specified to the environment variable as either the IP host name or the IP address. The Green Hills License Manager also understands two special host names:

LOCAL

(UNIX only) Attempt a connection over a local (non-network) communications path. You must use this in conjunction with the **-n** option when launching the licence server. On UNIX, native System V IPC is used instead of UDP.

BROADCAST

Search for the license server with a Server Address Resolution broadcast message. This has the same effect as the default.

There are also two ways to specify the value for this variable:

Search

Broadcast to all hosts specified in parallel [default]. A successful reply from any of them results in a connect. This method is recommended and is generally the faster. For example, **GHS_LMHOST=server1:server2** tells the tools to use the license server running on either the machine named *server1* or *server2*, or any machine reachable with a UDP broadcast.

Absolute (@)

If the host list is preceded by an at (@) sign, then the client attempts to connect to each host in order until the time-out (default 15 seconds). Use this method when you want to connect to a specific host. For example, **GHS_LMHOST=@server1,server2** tells the tools to use the license server running on the machine named *server1*, and to use the server running on *server2* if *server1* is not available.

Using the Command Line

Green Hills License Manager commands can be run from the command line in a standard format:

```
command -a -b -c -d argument -e argument
command /a /b /c /d argument /e argument
command /a /b /c /d argument /e:argument
```

You can use either a minus sign (-) or slash (/) to indicate an option. The colon (:), or space between the option letter and argument is not required.

Option letters are single characters. Unless otherwise specified, you can omit the space between the option letter and argument.

Green Hills License Manager commands conform to the method of invocation of the local operating system-with some additional flexibility.

Arguments must follow their associated options.

Getting Help

All Green Hills License Manager commands take the **-?** option, which displays a brief command usage and a summary of associated optional flags.

Retrieving Debug Output

You can retrieve debugging information for both the client application and the license server.

Debugging the Client

For your client application, you can turn on a limited form of debugging by setting the value of the client's environment variable **ELMDEBUG** to non-zero.

```
ELMDEBUG=1; export ELMDEBUG
```

You can use this debugging set-up to help you diagnose problems, especially those related to client application's communication with the license server.

By default, debugging information for client applications is sent to the standard output. If you follow the debugging level with a comma and a file name, then debugging information is sent to the named file.

```
set ELMDEBUG=1
```

or

```
set ELMDEBUG=1,file
```

where file is an absolute pathname of a file to receive the debugging messages.

For example,

Unix bourne-type shells:

```
ELMDEBUG=1,/tmp/elmdebug.out; export ELMDEBUG
```

Unix csh-type shells:

```
setenv ELMDEBUG 1,/tmp/elmdeland.out
```

Debugging the License Server

The license server debugging output is sent to the license server log file. If you want the license server to generate debug information, increase the verbosity level using the **-v** option of the **ghs_elmd** command.

For more information, see "ghs_elmd" on page 28.

Administrator Commands

This chapter contains:

- Administrator commands

This chapter describes the Green Hills License Manager administrator commands available with your application. The commands are:

- **elmadmin/xelmadmin** -- Invokes a license administrator program. Its functions include installing license keys, listing the users of licensed products, as well as other administrative functions.
- **elmalert** -- Provides notification when licenses become available or are close to expiration. The command checks the availability or expiration of licenses and either sends mail or prints a message to the screen with the requested information.
- **ghs_elmd** -- Starts the Green Hills License Manager license server. Your system should run the license server in your system start-up script.
- **elmrpt** -- Produces a report of licenses available or license activity from one or more license server log files.
- **elmusage** -- Reports a *snapshot* of current license activity and prints to standard out, updated every ten seconds.
- **elmver** -- Scans specified files for the Green Hills License Manager version information.

elmadmin

The **elmadmin** command invokes a license administrator program. Its functions include installing license keys, listing the users of licensed products, as well as other administrative functions.

The **xelmadmin** command is the X-11/Motif GUI version of **elmadmin**.

Usage

```
elmadmin [ -b cid -c -C file -d -e keydir -h -i -k -l -p -q -r n -s -v n  
-V C -x -z ] [features | alias | feature=alias,... ]
```

Options

-b *cid*

Buries the client by returning all licenses used by the client whose client ID is *cid* and marking this client as dead. You can display the client ID (CID) by entering **elmadmin -l**.

-c

Installs a license key and prints a server code used to create a license key. A single application may have one or more keys. The default directory is `/usr/green/key`. You can change the key directory using the **-e**

option. For more information on creating license files, see “Example 1: Installing a License Key” on page 21.

-C *file*

Installs a collection file (*file*) in the specified key directory (or the default key directory if you do not use the **-e** option) and signals the license server to read the newly installed collection files. You must create a collection file and store it in your key directory to use this option.

-d *

Sends a command to the license server to send the contents of the license server database to the log file. This option is most commonly used for developer testing and debugging.

-e *keydir*

Places the license keys in directory *keydir*. You can use this option with the **-c** option. Use this option to specify a key directory other than the default.(/usr/green/key)

For example, the following command line creates the license key in the directory **/local/my_app**.

```
elmadmin e /local/my_app c
```

This directory must already exist; the **elmadmin** command does not create it.

-h

When used with **-l**, produces an expanded listing to include each client's hold time and the shared license count.

The hold time is the duration that the licenses will be held for the user after the client exits. If the user starts the application again before the hold period ends, the held license is assigned to the user. The hold time amount is specified in the resource file. See Appendix A, “Resource Files”. If the client is held, column one of the listing is flagged with an H.

The shared license count is the number of clients that are currently sharing the license for the indicated feature. If the client has one or more shared licenses, column one of the listing is flagged with an S.

-i

Prints the release number of the **elmadmin** command and the version of the Green Hills License Manager Application Library with which it was compiled.

- k ***
Kills the license server. The license server terminates gracefully.
- l**
Lists the current users and outstanding licenses available for all features known to all license servers on the network. Known features are those features used since the license server was started, usually the last time the system was rebooted. If one or more features are on the command line, the listing contains the named feature(s) only. See “Example 3: Listing User and License Information” on page 23. This option lists license packs individually.
The **-l** option scans all license servers on the network.
- p**
Prevents ping. Use this option with the **-c** option. Using the **-p** option, the license server is not notified that a new key has been installed. This option is useful when you are installing a large number of keys at once.
- q**
Queries and prints environment information for debugging. The query output includes the current host name and its network address, the current port and license server name, GHS ID, and host code.
- rn**
-rn/key
Sets ready-key mode for use with the **-c** option. The *n* argument is the number of license servers that will be running (usually one, unless redundant servers are being used). If *n* is positive, the command does not print the server code(s). Instead, it prompts for the server key and displays information associated with the key. If *n* is negative, the command does not prompt for the server key. Instead, it displays the specified number of server codes. See “Example 2: Setting Ready-Key Mode” on page 22.
- s**
Finds all license servers using *server address resolution* and prints their names. If a feature name appears on the command line, the hosts responding with this feature are printed.
- v n ***
Changes the license server log file verbosity level to *n*. You can use this option to change the level after the license server has started.
- x**
Inhibits the copyright notice.

-z *

Zeroes out the log file. For example, the following command lines can be used to clean up a log file every evening on a UNIX system:

```
ELMHOST=@localhost
export ELMHOST
elmadmin -z
```

(*localhost* can be a different host if you want to access **ghs_elmd** remotely.) Alternatively, the limit log file size option of the license server may be used to automatically keep the log file under control.

Truncating the log file by other means would be ineffective because the license server keeps the log file open at all times. The next license server log file write uses the old file pointer, filling the file to the original length. Thus you must use the **-z** option or the log file size option of **ghs_elmd** to truncate the file.

* Can be invoked only by the owner of the current license server process, the *super-user*, or a member of group *wheel*.

Example 1: Installing a License Key

The following is an example of using the **elmadmin** command to install a license key for feature 99 in the default directory:

```
elmadmin -c
On how many hosts will you run the license servers?
  (default=1):1
Code for "opus.bloomcnty.com" is: 64/7809 4524 9869 048
Please enter your key: 1234 5678 9102 8765 4321
Feature Name: 99
Number of licenses: 3
Start date:
Expiration date: Sun Mar 31 23:59:59 (17.09 days from now)
Successfully installed key: /usr/green/keys/99.lic
Notifying license server of new key...OK
```

If an *alias* appears as the final argument, it is installed as this feature's alias. For example, if you entered the command,

```
elmadmin -c ezchart
```

Green Hills License Manager uses *ezchart* as the alias for feature 99. Commands such as **elmrpt** print this alias instead of the feature number.

A list of definitions may also appear on the command line. Upon decoding the key input, the feature name gleaned from the key will be matched against the list of definitions. If a match is found, the name following the equal sign (=) is taken as the feature alias. This provides a convenient method of specifying feature aliases within shell scripts. For example, the following command selects the alias *xrunner* if the key decodes with the feature 01.

```
elmadmin -c 01=xrunner,02=xprofile,03=xdebug
```

Example 2: Setting Ready-Key Mode

To set ready-key mode, use the **-c** option with the **-r n** option. If *n* is positive, **elmadmin** does not print the server code(s). Instead, it prompts for the server key and displays information associated with the key.

```
elmadmin -c -r1
Please enter your key: 1234 5678 9102 8765 4321
Feature Name: 99
Number of licenses: 3
Start date (day-mon-yr or <return> for none):
Expiration date (day-mon-yr, # days from now, or <return>
for none): Sun Mar 31 23:59:59 (17.09 days from now)
Successfully installed key: /usr/green/keys/99.lic
Notifying license server of new key...OK
```

If *n* is negative, **elmadmin** does not prompt for the server key. Instead, it displays only the specified number of server codes.

```
elmadmin -c -r-1
Code for "snoopy.doghouse.com" is: 64/7809 4524 9869 048
```

If a slash and a license key follow the license server count, **elmadmin** displays the information associated with the key. (You can use single quotes instead of double quotes):

```
elmadmin -c -r1/"1234 5678 9102 8765 4321"
Feature Name: 99
Number of licenses: 3
Start date:
Expiration date: Sun Mar 31 23:59:59 (17.09 days from now)
Successfully installed key: /usr/green/keys/99.lic
Notifying license server of new key...OK
```

Example 3: Listing User and License Information

The following is an example of a listing generated with the **elmadmin -l** command:

```
CIDLID  PACKUser          Feature  Group  Started
D 1 1 02.0 young@qback  football offense Sep 2 16:03
S 3 2 02.0 elway@qback  football offense Sep 2 13:35
S 4 2 02.0 marino@qback football offense Sep 2 11:36
H 5 3      aikman@qback football offense Sep 2 12:34
football [98], pack 02.0: 50 licenses, 4 in use; installed Sep-1-96.
Expires Jan-26-97.
```

The code in the first column describes the status of the client.

- H** Indicates that this client is a *held* client. The license server holds a license for a specified time after the held client exits. See Appendix A, “Resource Files” You can also list the hold period using the **-h** option.
- D** Indicates that this client is connected but disabled due to too few license servers currently running in redundant license servers mode.
- S** Indicates that the client has a *shared* license for the indicated feature. Shared license groups share the same LID.

The CID column lists the client ID. Each client application is assigned a unique ID. You can use the CID with the **-b** option.

The LID column lists the license ID. These are unique license numbers. Each client usually has a unique LID for each licensed feature, except when the license is shared. If clients share a license, they share the LID (see users *elway* and *marino* in the example).

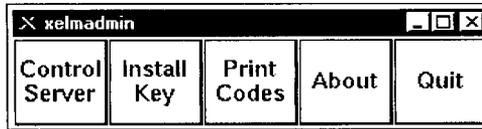
The user is displayed as *user@host,display*, where *display* is the value of the X-11 DISPLAY when the application was invoked. The *display* may be missing for non X-11 applications.

xelmadmin

The **xelmadmin** command is the X-11/Motif GUI version of the **elmadmin** command. Like **elmadmin**, **xelmadmin** is a license administrator program for use with applications embedded with Green Hills License Manager. Its functions include installing license keys, listing the users of licensed products, and other Green Hills License Manager administrative functions.

Window Manager borders differ from one system to another. All examples displayed in this chapter were generated under *twm*, the Tab Window

Manager. When **xelmadmin** is invoked, it displays a button bar which summarizes the available options:



Syntax

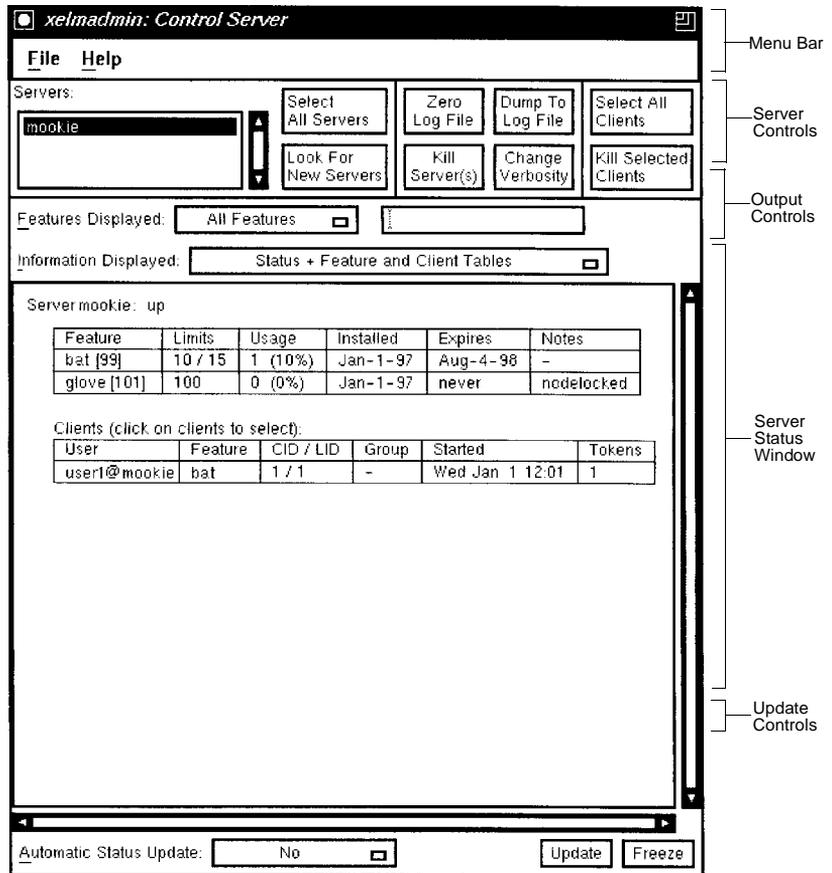
xelmadmin [-i -q]

Options

- i Prints the release number of **xelmadmin** and the version of the Green Hills License Manager library with which it was compiled.
- q Queries and prints various environment information for debugging. Output includes the current host name, its network address, the name of the **GHS_LMHOST** environment variable and its current setting, the current port and license service name, the Vendor ID, and the *hostcode* value.

Control Server

Click **Control Server** to access the **Control Server** dialog box. Using this dialog box, you can monitor and control server and license usage as well as signal one or more servers to perform various actions. The **Control Server** dialog box is composed of five elements: menu bar, server controls, output controls, server status window, and update controls.



The following list describes each of the five components.

Menu Bar

The menu bar enables you to dismiss the window, quit the application.

Server Controls

The Servers window displays the list of servers which were located using broadcast (or were specified by an @-prefixed **GHS_LMHOST** environment variable setting). You can select any subset of the servers by pressing CTRL and clicking the names of servers on the list. You can also click **Select All Servers**. To find new servers using broadcast, click **Look For New Servers**.

The additional server control options enable you to signal the selected servers to perform the following:

- **Zero Log Files**—Zero out log files.
- **Dump To Log File**—Dump debugging information to servers' log files.
- **Select All Clients**—Select all clients displayed in the Server Status window.
- **Kill Server(s)**—Terminate the selected servers.
- **Change Verbosity**—Change the server's verbosity level.
- **Kill Selected Clients**—Kill the selected clients.

Output Controls and Server Status window

The output controls enable you to select the type of server information you want to display in the Server Status window. The following options control the output.

- **Features Displayed**—Sets the display for all features or one or more specified features. To specify a set of features, set this option to **Features Specified** and enter the name or alias of each feature in the box to the right. Separate the features in the list with commas.
- **Information Displayed**—Sets the type of feature status displayed.

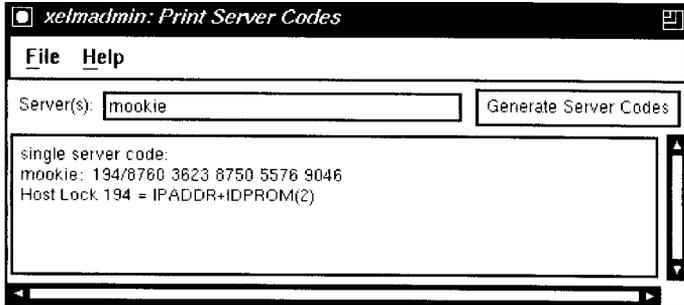
Update Controls

The update controls enable you to control the update of the status information.

- **Automatic Status Update**—Sets a time interval for automatic updates of the status.
- **Update**—Requests a status update.
- **Freeze**—Freezes/continues automatic updating.

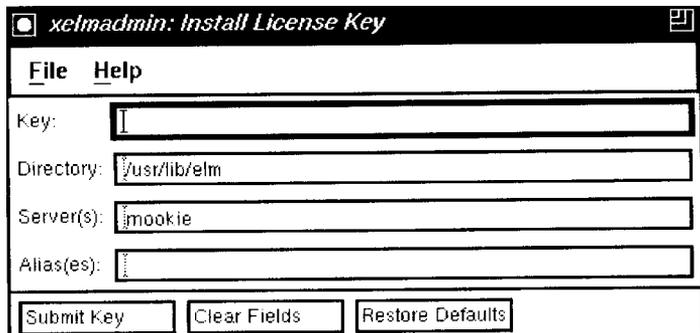
Print Codes

Click **Print Codes** to display the server code for the machine on which **xelmadmin** is running. On the **Print Server Codes** dialog box, you can generate server codes for one or more other servers by entering a comma-separated list of server names in the **Server(s)** box and pressing ENTER or by clicking **Generate Server Codes**.



Install Key

Click **Install Key** to install a key on the specified servers in the stipulated key directory. On the **Install License Key** dialog box, specify multiple servers with a comma-separated list. You can specify a single alias for the feature in the **Alias(es)** box or enter a comma-separated list in the format *feature=alias*. In the latter case, if the key's feature matches the feature in one of the pairs listed, it will be installed with the corresponding alias. For example, if you enter `01=green, 102=red` in the **Alias(es)** box and the key is for feature 102, then feature 102 is installed with the alias *red*.



To install a key and signal the server to read the new key file, enter the key information and click **Submit Key**.

About

About displays information about **xelmadmin**.

Quit

Quit exits **xelmadmin**.

Related Commands

elmadmin, **ghs_elmd**

ghs_elmd

The command **ghs_elmd** starts the Green Hills License Manager license server. Your application should execute the license server from your system start-up script, usually one of the following:

```
/etc/rc.local  
/etc/inittab  
/etc/rc2.d  
/etc/rc.local
```

The license server must be running for any application licensed with Green Hills License Manager to run.

When running, the license server processes client requests. This includes issuing and returning licenses and interfacing with the **elmadmin** command to list the current users of licensed applications.

For each licensed application, Green Hills License Manager looks for a key file in the key directory. Most often, the key directory or directories is specified with the **-e** option. Key files may be freely installed or updated using **elmadmin** or **xelmadmin** while the license server is running. Keys will automatically be reread if the license server detects a change in the modification date.

Syntax

```
ghs_elmd [-D -e path -f -i -l file -m size -n -p n -r file -s n -v n -x -z n]
```

Options

-D

Activates debug mode and flushes the log file after each message. Normally the log file is flushed after each packet is processed. (There can be many messages per packet, depending on the verbosity (**-v**) level.) This option also allows **sigquit** to take the default action of creating a core file and exiting. In addition, in background mode, the directory does not change. Normally, **ghs_elmd** runs with the current working directory set to “/.”

The message “DEBUG mode is on” prints to the log file if you use debug mode. See “-v n *” on page 20 for changing the verbosity level of **ghs_elmd**.

-e path

Loads all keys found in the specified key directory or set of directories. If none is specified, the default key directory is used. Multiple directories may be specified as a colon-separated list of directory names. For example.

```
ghs_elmd -e /local/elan:/local/express:/usr/lib/elm
```

-f

Runs the license server in the foreground. Usually run in the background, the license server forks a child process and detaches from the terminal, and the parent exits. The **-f** option inhibits this process. You can, for example, run **ghs_elmd** from **inittab**, instead of **/etc/rc.local** or **/etc/rc2.d**, with the **-f** option.

-i

Prints the Green Hills License Manager release number, the vendor ID, and exits. The vendor ID provides a way to verify that the same vendor salt is being used among all the Green Hills License Manager commands. The vendor ID should be the same for **ghs_elmd** and **elmadmin**.

-l file

Creates a log file and writes all relevant information to that file. If the file already exists, this option appends the new information to the end of the file. The specified file name must be a complete path name.

Using a log file is highly recommended because it provides a way of tracking problems should they occur. You must have a log file in order to use **elmrpt** or **xelmrpt**.

Trim your log file if you are using one. You can trim the file automatically by using the **-m** option or using a nightly command spawned from *crontab*. The following pair of commands is an example of how to keep the log file under control with the nightly *crontab* scheme:

```
cp /usr/adm/elm.log /usr/adm/elm.log.old
ELMHOST=@somehost elmadmin -z
```

where *somehost* is the host name where the license server is running.

-m *size*

Sets the size limit for the log file. With this option, the log file is self-maintained by the license server, such that the license server limits the size to which the log file may grow. When this size is reached, the log file is moved to *file.old* and the current log file is truncated. The size is an integer or floating point number. The default unit is in bytes. It may also be specified in kilobytes or megabytes by suffixing a *k* or an *m* to the size, respectively. For efficiency, the log file size is actually checked only every 100 lines, so it may grow beyond the limit slightly before it is backed up and truncated.

-n

Forces **ghs_elmd** to use System V native inter-process communication. This option is used for local communication for systems without a network. It is applicable on small UNIX systems such as SCO and Interactive UNIX without TCP/IP installed.

When **-n** is used with **ghs_elmd**, you must set **GHS_LMHOST** to **LOCAL**.

On System V, the default inter-process communication queue is small, adequate for about two client processes talking to the license server concurrently. The license server attempts to increase the queue size, but since only root can raise the IPC queue size, **ghs_elmd** must be run as *super-user* to be effective.

-p *n*

Uses the port address *n* instead of the default or one specified by the system services file. For example, you can use this option if there was a conflict in port addresses and the user does not have permission to alter the system port services file.

-r *file*

Applies the specifications in the resource file, see Appendix A, “Resource Files”. If this file is changed, the license server automatically rereads it. It is not necessary to restart **ghs_elmd**.

-s *n*

Sets the start-up time to *n* seconds. This is the period that the license server waits and listens for clients reconnecting after a system crash. The time period gives existing clients a chance to reconnect before new clients can connect. The default is usually 180 seconds (3 minutes). You can customize the default in the salt file.

-v *n*

Sets the verbosity of the messages printed to the log file. When a level is selected, all levels less than or equal to the level are printed. Levels five and higher are considered debugging levels. The license server defaults to level three. The levels are:

- 1-Only error messages.
- 2-License failures.
- 3-License activity.
- 4-Client connects and disconnects.
- 5-Message for every packet received.
- 6-Message for every packet sent.
- 7-More client and zombie information.
- 8-Information about key and resource file.
- 9-Maximum amount of information.

-x

Turns on log file tag. If log file tags are not already turned on by default, this turns on log file tagging (secure checksum marks).

-z *n*

Sets the *zombie* period to *n* seconds. If a client does not communicate with the license server within this time period, the license server assumes the client is dead and retrieves its licenses. The default zombie interval is usually 3 minutes. The license server cleans up zombies every 60 seconds, so it may take up to 1 minute more for the licenses to be returned.

The zombie time-out must be greater than or equal to the value set by the application developer.

Examples

The following is an example of using **ghs_elmd** in the system start-up file **/etc/rc.local**. It sets the key directory to **/local/mercury/keys**, the log file to **/local/mercury/logs/ghs_elmd.log**, and limits the log file size to 2.5 Megabytes:

```
ELMD=/local/mercury/bin/ghs_elmd
ELMDKEYS=/local/mercury/keys
ELMDLOG=/local/mercury/logs/ghs_elmd.log
if [-f $ELMD ]; then
    echo "`$ELMD -i` started." > /dev/console
```

```
$ELMD -e $ELMDKEYS -l $ELMDLOG -m 2.5m > /dev/console  
fi
```

Related Commands

elmrpt, elmadmin, ghs_elmd

elmver

The **elmver** command scans specified files for the Green Hills License Manager version information. For example, if you wanted version information for **elmadmin**, use the following command line in the **commands** directory:

```
elmver elmadmin
```

The output is similar to the following:

```
elmadmin: Elan License Manager release 5.0.1040
```

Use the **-v** option to generate information on signal-handling function, how the ID Prom is determined, and if the ethernet address is available. For example,

```
elmver elmadmin -v
```

results in the following verbose output.

```
file: Elan License Manager release 5.0.1040  
file: HAVE_SIG=SIGACTION  
file: HAVE_IDPROM=GETHOSTID  
file: HAVE_ETHER=TRUE
```

Files with version information are those that

- are compiled with the Green Hills License Manager Application Library.
- perform license server communications.
- The **elmver** command is convenient for determining the release level of the Green Hills License Manager library with which a command was compiled.

Syntax

```
elmver [-v] files...
```

Options

-v

Sets verbose mode, displaying additional version information.

A

Resource Files

This appendix contains:

- Restricting the domain
- Changing default connection settings
- Listing redundant servers
- Reserving, excluding, and holding licenses
- Examples

The resource file contains Green Hills License Manager related resource information for one or more features. You can use resource files supplied by Green Hills or create your own. The files can specify the following:

- An address mask to filter out requests from foreign systems (domain licensing)
- Reserved licenses (licenses that are reserved for individuals, groups, or machines)
- Excluded licenses (licenses that exclude specified individuals, groups, or machines)
- Held license periods (minimum period for which a license is held for a user, group, or machine)
- A list of redundant servers

Each line of the resource file contains a single definition. A number sign (#) at the beginning of any line is treated as a comment (the remainder of the line is ignored).

To use the resource file, the license server must be started with the resource file option (**ghs_elmd -r**).

Restricting the Domain

By default, the Green Hills License Manager accepts requests from any IP address. If desired, requests may be limited to only those addresses that conform to your specifications. For example, if your company owns a Class-A network, but is also on the Internet, you may want to limit license server requests from only those on your Class-A network domain.

Packet requests may be limited to machines defined by an IPACCEPT mask specified in the resource file. Specify the *mask* in the resource file as a line of the form

```
%IPACCEPT mask
```

where *mask* consists of four octets, separated by periods. Each octet may be one of the expressions in the chart below. An octet can also consist of a comma-separated list of expressions.

Expression	Description
N	An integer between 0 and 255.
N-M	An integer range in which both <i>N</i> and <i>M</i> are between 0 and 255. <i>N</i> is always less than <i>M</i> . For example, 200-255
*	An asterisk matches all integers.

Expression	Description
THIS	<p>The current license server's IP octet value is inserted in the position designated by "THIS".</p> <p>For example, if the host address is "192.100.42.16,"then "THIS.*.*" is the same as "192.*.*" and "THIS.THIS.THIS.*" is the same as "192.100.42.*"</p> <p>As another example, the resource line, %IPACCEPT THIS.*.* limits requests to only those clients on the same Class-A network as the license server. Whereas, the resource line, %IPACCEPT THIS.100,200-224,226.*.* limits requests to only those clients which are on the same Class-A network and are from Class-B sub-networks 100, 200 through 224, and 226.</p>

Changing Default Connection Settings

Two configuration parameters dealing with connecting to the server are ELMRETRIES and ELMTIMEOUT. ELMRETRIES defines the maximum number of reconnection attempts to the server. The total number of tries is (ELMRETRIES+1). ELMTIMEOUT determines the total time for all retries before the connection times out. The time for each try is ELMTIMEOUT divided by (ELMRETRIES+1). The default settings are:

For Unix tcsh or csh:

```
setenv ELMRETRIES 2
setenv ELMTIMEOUT 15
```

For Unix sh:

```
ELMRETRIES=2
export ELMRETRIES
ELMTIMEOUT=15
export ELMTIMEOUT
```

This default setup makes three attempts to connect, one attempt every five seconds. If you are having network problems, Green Hills recommends that you first try changing the time out before trying to change the retries.

Listing Redundant Servers

Green Hills License Manager can be run with one or more license servers to implement license server redundancy (backup license servers). The license server reads the names of the other redundant servers from the key files in the

key directory. So, you should specify the key directory when you launch the license server.

However, if you do not specify the key directory when you launch the license server, the key files are not preloaded and the license server does not know the names of other redundant servers. In this case, the host name of each redundant license server should be included in the Green Hills License Manager resource file.

The server host names are listed in the form

```
%SERVER hostname
```

where *hostname* is the name of one of the redundant hosts. The %SERVER definitions are required only for use with redundant servers.

Reserving, Excluding, and Holding Licenses

Feature licenses can be *reserved* for individual users or client machines using a resource file. Individual users and/or client machines can also be excluded from using specified features. Reserved and excluded licenses are indicated in the resource file using the format

```
feature:group:client1,...,clientn:k:h
```

The maximum number of characters per line is 512. The following describes the items of each line:

feature Feature name or alias, a maximum of 32 characters.

group

Group name, a maximum of 20 characters. This is not related to the UNIX system group. It is the name you want to call this group of clients. This group name will be displayed in a listing generated by the **elmadmin -l** command.

client

User login names or host names if preceded by the at sign (@). For example, for the client *godzilla@tokyo*, *godzilla* is a user name, and *@tokyo* is a host name.

Client names are separated by commas. An empty group list means everyone. This is useful for setting the hold period only. (Often used with *k* also set to zero.)

k

Number of licenses reserved for the group or the word EXCLUDE to exclude this group from using *feature*. Lines for excluding licenses must appear before reserved lines in the resource file. To specify a hold time for *group* only, use zero. Use an empty *group* and zero *k* to set a default hold period for everyone. The hold time is ignored for lines excluding licenses.

h

Hold period, in seconds, for all members of this group. When a client terminates, all currently licensed features will be held for this many seconds. If the same user on the same host runs the application again, these held licenses are granted to that user. Thus, held licenses are a form of reserving licenses for active users.

The hold period is determined from the resource file as follows: If the license requested comes from the reserved license pool for a group, it has the group's hold period. If the license requested comes from the free license pool, the first line that matches the feature and group member will contain the hold period that is used. Therefore, the order in the resource file may be important. (See the example.)

Examples

The following is a sample resource file:

```
# Accept requests from this Class-C network only.
%IPACCEPT THIS.THIS.THIS.*
# We're using 3 redundant host servers.
%SERVER mars
%SERVER jupiter
%SERVER saturn
# Our reserved and excluded licenses...
99:mother-in-law:endora,@cauldron:EXCLUDE:0
1045:wordproc:john,paul,george,ringo:3:600
cyberpunk:hackers:gibson,stephenson,simmons:3:120
poetsys:lab:@byron.mars.com,@shelley.mars.com:1:0
# Everyone else using 99 has default hold period of 60 sec
99:default::0:60
```

In this resource file,

- An IPACCEPT mask specifies that requests should be accepted from the local class-C network only.
- Three host servers have been defined—*mars*, *jupiter*, *saturn*—because redundant license servers are in use. If redundant license servers were not used, these lines would be unnecessary.
- The user *endora* or anyone logged onto the machine *cauldron* would be excluded from checking out any licenses for feature 99.
- Three licenses for feature 1045 have been reserved for four members of the word processing group, *wordproc*, whose login names are *john*, *paul*, *george*, and *ringo*.
- Three licenses of *cyberpunk* have been reserved for the member of the group *hackers*.
- One license for *poetsys* has been reserved for anyone on the machines *byron.mars.com* or *shelley.mars.com*.
- The user *endora* or anyone logged onto the machine *cauldron* does not have a hold period. Users *john*, *paul*, *george*, and *ringo* have a hold period of 600 seconds. Users *gibson*, *stephenson*, and *simmons* have a hold period of 120 seconds. Anyone on the computers *byron.mars.com* or *shelley.mars.com* has no hold period, and everyone else has a 60-second hold period by default.

Lines with EXCLUDE must precede lines for reserved licenses. If both host names and user names are used, the first match of host or user name, from left to right and top to bottom is used. A default hold period should be indicated last. You can establish a minimum hold period within a client, which might be greater than the value specified in the resource file.

B

Troubleshooting

This appendix contains:

- What to do if the license server is not working
- What to do if you experience delays
- Other causes of problems

What to do if the Elan license server is not working

1. Check to see if the license server is running

For a node-locked setup:

- It may be worthwhile to check for a license manager (via the commands below) even if you know you have node-locked licenses.
- The tools should take a node-locked license in preference to a floating license, but if you are experiencing problems with licensing, investigating the presence of floating licenses may shed some light on your situation.
- If a node-lock license is unavailable, expired, or incorrectly installed, the tools will realize this, and may try to obtain licenses from the network. This could disguise a problem with your node-locked license:
 - For example, if your node-locked license is invalid, a tool may try to contact a network license server and be told there are no more licenses. It would print out a message to the effect that there are no more licenses, (which is true) but the solution to your problem is probably to reinstall your node-locked license.
 - For these reasons, knowing that license servers are running can help when diagnosing license problems.

On the machine where the license manager is supposed to be running:

- **For Solaris2 and other SysV unix:**

```
$ ps -fe | grep ghs_elmd | grep -v grep
root 26696      1  0 18:01:27 ?          0:00 ghs_elmd
$
```

- **For BSD style unix:**

```
$ ps auxww | grep ghs_elmd | grep -v grep
root      1392  0.0  1.4 116 396 ?  S   18:33   0:00
./ghs_elmd
$
```

- **For Windows:**

Check the Task Manager for a process called ghs_elmd. To check the status and reachability of remote license managers on the network:

```
$ elmadmin -s
GHS License Manager - Copyright 1989-1997 Elan Computer
Group, Inc.
```

License Server Host	Status
server	Active
cathy	Active
dogbert	Active
lutra	Active
cyrano.ghs.com	Active
runner	Active
islay.ghs.com	Active
mud.ghs.com	Active
newport	Active
beavis	Active
illuminati.ghs.com	Active
ender.ghs.com	Active
\$	

2. Check that the correct licenses are installed

- **elmusage**

elmusage respects GHS_LMHOST, so if you want to check the usage of license on the host lm1, then set GHS_LMHOST to "@lm1" and run elmusage.

- **elmadmin -l**

elmadmin will ignore GHS_LMHOST, but will report significant detail about the license in use and available on all reachable license servers.

- For node-locked licenses, you will have to look in the nodes directory created when you installed the distribution:
 - Each license will be installed in a file named after its feature number. For feature 100 (multi) you would look for a file called "100.lic".
 - If you want to know which feature a license file corresponds to, you can look at the license file (be sure not to make changes to it, though.) There will be a line with the ascii name of the product just below the first line of numbers. This line should be present in any key produce by GHS which was installed correctly. If the line is not there, it doesn't mean that the key is invalid, but it is a reason to be suspicious (maybe the key should be reinstalled.)

3. Enable license manager diagnostics and look for unusual messages

1. First, set ELMDEBUG so that the diagnostics will be generated and directed to a file.

```
setenv ELMDEBUG 1,file
```

2. Next, run the command that is experiencing license manager failures. If it is a long-running command (e.g. a GUI command) then you probably want to quit it soon to limit the number of diagnostic messages.

```
ccomppc hello.c
```

3. Look in the file named in the ELMDEBUG variable. Here is an annotated example of a correctly operating program obtaining a floating license.

```
98-Dec-17 21:25:09: GHS License Manager 5.0.6 [VID ab345d] on UNIX
98-Dec-17 21:25:09: elm_connect(0, "", x0.0.0.0, "2901",
"/usr/green/keys", x0, 0, n)
```

`/usr/green/keys` is a default, and is nothing to worry about even if your keys are installed in a different directory, or if there is no license manager on this host.

```
98-Dec-17 21:25:09: ELMTIMEOUT = 15 seconds, ELMRETRIES = 2
```

ELMRETRIES (an environment variable) specifies how many time the license server will be retried (total of ELMRETRIES+1 attempts.)

ELMTIMEOUT is the total timeout for all attempts. The per-attempt time-out is (ELMTIMEOUT/(ELMRETRIES+1)) seconds.

```
98-Dec-17 21:25:09: broadcast_to_host "localhost"
98-Dec-17 21:25:09: Can't broadcast to "/usr/green/keys/2901.lic"
hosts: Unable to open file
```

This file not found is a check for node-locked licenses. If you do not have any, then this is an expected message.

```
98-Dec-17 21:25:09: broadcasting ...
98-Dec-17 21:25:09: Found 2 nets.
98-Dec-17 21:25:09: Name="lo0"
98-Dec-17 21:25:09: loopback
98-Dec-17 21:25:09: Name="hme0"
98-Dec-17 21:25:09: sending to 192.168.2.255
98-Dec-17 21:25:09: SAR timeout=15
98-Dec-17 21:25:09: ELMTIMEOUT = 15 seconds, ELMRETRIES = 2
98-Dec-17 21:25:09: RECEIVED PACKET from server
98-Dec-17 21:25:09: server: YES_LICENSES(268435455); cut timeout to 2
seconds.
98-Dec-17 21:25:09: sending packet #2 RQ I
98-Dec-17 21:25:09: Receive: Wait 5 sec for packet...
98-Dec-17 21:25:09: Packet #2 I RP K received.
98-Dec-17 21:25:09: get_peers 1 servers
98-Dec-17 21:25:09: server model set to single
98-Dec-17 21:25:09: sending packet #3 RQ V
98-Dec-17 21:25:09: Receive: Wait 0 sec for packet...
98-Dec-17 21:25:09: Alive: Dead daemon
```

```
98-Dec-17 21:25:09: resending packet #3 RQ V
98-Dec-17 21:25:09: sending packet #4 RQ G
98-Dec-17 21:25:09: Receive: Wait 5 sec for packet...
98-Dec-17 21:25:09: Packet #3 V RP K received.
98-Dec-17 21:25:09: Receive: Wait 5 sec for packet...
98-Dec-17 21:25:09: Packet #3 V RP K received.
98-Dec-17 21:25:09: Receive: Wait 5 sec for packet...
98-Dec-17 21:25:09: Packet #4 G RP K received.
98-Dec-17 21:25:09: sending packet #5 RQ U
98-Dec-17 21:25:09: Receive: Wait 5 sec for packet...
98-Dec-17 21:25:09: Packet #5 U RP K received.
98-Dec-17 21:25:09: sending packet #6 RQ p
98-Dec-17 21:25:09: Receive: Wait 5 sec for packet...
98-Dec-17 21:25:09: Packet #6 p RP K received.
98-Dec-17 21:25:09: sending packet #7 RQ i
98-Dec-17 21:25:09: Receive: Wait 5 sec for packet...
98-Dec-17 21:25:09: Packet #7 i RP K received.
98-Dec-17 21:25:10: sending packet #8 RQ B
98-Dec-17 21:25:10: Receive: Wait 2 sec for packet...
98-Dec-17 21:25:10: Packet #8 B RP K received.
```

And that's the end of the program.

4. Look at the license server's log

Here's a few highlights from a log. These highlights point out some of the normal things which you can ignore when debugging.

```
98-Dec-17 21:41:53: GHS License Manager 5.0.6 [VID ab345d] on UNIX
98-Dec-17 21:41:53: slm_hostid: no VENDORHOSTID
98-Dec-17 21:41:53: slm_hostid: no dongle serial number: Security
devices are not enabled
98-Dec-17 21:41:53: slm_hostid: gethostname returns "clienthost"
98-Dec-17 21:41:53: slm_hostid: HOST="clienthost"
98-Dec-17 21:41:53: slm_hostid: IDPROM="80080110"
98-Dec-17 21:41:53: slm_hostid: HOSTID="clienthost80080110"
98-Dec-17 21:41:53: slm_hostid: gethostname returns "clienthost"
98-Dec-17 21:41:53: slm_hostid: HOST="clienthost"
98-Dec-17 21:41:53: slm_hostid: slm_ether failed: Bad file number
98-Dec-17 21:41:53: slm_hostid: gethostname returns "clienthost"
98-Dec-17 21:41:53: slm_hostid: HOST="clienthost"
98-Dec-17 21:41:53: slm_hostid: no DISK ID
98-Dec-17 21:41:53: slm_hostid: no DISK ID
98-Dec-17 21:41:53: Warning: Unable to obtain HostID #1
(VENDORHOSTID) on "clienthost": Error.
98-Dec-17 21:41:53: Warning: Unable to obtain HostID #2
(DONGLESERIALNUM) on "clienthost": Security devices are not enabled.
98-Dec-17 21:41:53: Warning: Unable to obtain HostID #4
(HOSTNAME+ETHER) on "clienthost": Error.
98-Dec-17 21:41:53: Warning: Unable to obtain HostID #5
(HOSTNAME+DISKID(8)) on "clienthost": Error.
98-Dec-17 21:41:53: Warning: Unable to obtain HostID #6 (DISKID(8))
on "clienthost": Error.
```

Notice that all HostID's failed except #3. As long as one succeeds, you should be fine.


```
98-Dec-17 21:41:53: slm_hostid: HOSTID="clienthost80080110"
98-Dec-17 21:41:53: slm_hostid: gethostname returns "clienthost"
98-Dec-17 21:41:53: slm_hostid: HOST="clienthost"
98-Dec-17 21:41:53: slm_hostid: IDPROM="80080110"
98-Dec-17 21:41:53: slm_hostid: HOSTID="clienthost80080110"
98-Dec-17 21:41:53: Initialization period = 5 seconds.
98-Dec-17 21:41:53: Peer registered: "clienthost" clienthost80080110
c0a80217 192.168.2.23 192.168.2.23.
```

Here, we start reading license files and loading keys. At this time, Green Hill customers will not generally get a collection file (but one shows up in the listing below.) They usually get individual keys.

```
98-Dec-17 21:41:53: Exploding collection file
/usr/green/keys/clienthost.col.
98-Dec-17 21:41:53: Installing Pack 10 from collection file.
98-Dec-17 21:41:53: Loaded key /usr/green/keys/10.lic (maps) U tokens
[MINIKEY].
98-Dec-17 21:41:53: Loaded key /usr/green/keys/2201.101 (ccommp) U
tokens starts Aug 2 00:00:00 1998, EXPIRED Nov 1 23:59:59 1998
[STDKEY 062802369].
98-Dec-17 21:41:53: Loaded key /usr/green/keys/2901.101 (ccomppc) U
tokens [STDKEY 067224298].
98-Dec-17 21:41:53: Loaded key /usr/green/keys/809.105 (pbr1) 2
tokens expires Jan 1 23:59:59 2004 [STDKEY 903640701].
98-Dec-17 21:41:53: Loaded key /usr/green/keys/809.101 (pbr188) 2
tokens expires Jan 1 23:59:59 2000 [STDKEY 903639938].
98-Dec-17 21:41:53: Loaded key /usr/green/keys/809.102 (pbr189) 2
tokens expires Jan 1 23:59:59 2001 [STDKEY 903640504].
98-Dec-17 21:41:53: Loaded key /usr/green/keys/809.103 (pbr20) 2
tokens expires Jan 1 23:59:59 2002 [STDKEY 903640554].
98-Dec-17 21:41:53: Loaded key /usr/green/keys/809.104 (pbrboth) 2
tokens expires Jan 1 23:59:59 2003 [STDKEY 903640584].
98-Dec-17 21:41:53: Loaded key /usr/green/keys/809.106 (pbr1) 2
tokens expires Jan 1 23:59:59 2004 [STDKEY 903640740].
98-Dec-17 21:41:53: Loaded key /usr/green/keys/809.107 (pbrg189) 2
tokens expires Jan 1 23:59:59 2005 [STDKEY 903640778].
98-Dec-17 21:41:53: Loaded key /usr/green/keys/809.108 (pbrg1892) 2
tokens expires Jan 1 23:59:59 2005 [STDKEY 903640842].
98-Dec-17 21:41:53: Loaded key /usr/green/keys/809.109 (pbrg20) 2
tokens expires Jan 1 23:59:59 2005 [STDKEY 903640878].
98-Dec-17 21:41:53: Loaded key /usr/green/keys/809.110 (pbrgboth) 2
tokens expires Jan 1 23:59:59 2005 [STDKEY 903640922].
98-Dec-17 21:41:53: Loaded key /usr/green/keys/809.111 (pbrgboth2) 2
tokens expires Jan 1 23:59:59 2005 [STDKEY 903640983].
```

Now the license server starts to serve licenses. As you can see, there are two licenses granted and one returned.

```
98-Dec-17 21:43:04: joeuser@clienthost,clienthost:0.0 [1], pbr189
(809): LID 1 issued, 1 in use.
98-Dec-17 21:43:14: joeuser@clienthost,clienthost:0.0 [1], pbr189
(809): LID 1 returned, 0 in use (used: 0:00:00:10).
98-Dec-17 21:43:26: joeuser@clienthost,clienthost:0.0 [1], pbr189
(809): LID 1 issued, 1 in use.
```

Now the license server shuts down.

```
98-Dec-17 21:43:50: Received signal 15. GHS_ELMD Terminated.
```

5. Try reinstalling your keys

- First, follow the procedure you did when installing the keys the first time. You may see error messages stating that the keys are already installed. Ignore those for now.
- Now retry the tool. If the error has not gone away, then next you should try a complete reinstall of the keys:
 - To completely reinstall the keys, first make sure that you have the key files originally given to you by GHS for all of the keys that you want to keep on the machine. (These files are usually called "key_install.sh" for Unix, or "license.key" for Windows, however you may have given them other names.)
 - Next, examine your keys directory and/or your nodes directory (for you floating license and node-locked licenses respectively.)
 - For each key that you want to completely reinstall, delete the key file. For example, for multi, you would remove "100.1*".
 - Again, the key file contains the name of the product on its fifth or so line, so you can use that to distinguish the files.
 - If you want to reinstall all keys, then you can remove all of the key files. Now reinstall the keys as you did the first time (usually running "Install Licenses" from the Green Hills Licensing program group on Windows, or running the key_install.sh script on Unix.)

6. Make sure that the keys are not expired

You should be able to determine this from the output of `elmadmin -l` or from the license server log file. Generally, this should also be apparent from the output of the tools.

7. Read our manuals

8. Talk to tech support

Please include the information generated above. Be sure to also mention the host platform, (Win32, Solaris2, SunOS 4, ...) the version of the tools, (1.8.8 PPC, 1.8.9 MIPS, ...) the OS of the machine on which the license server is

running, and the type of licenses you have (floating or node-locked.). All of this information will help improve the response time.

If your license server is reporting that there is something wrong with your licenses, then it might be helpful to include a tar or zip archive of your "keys" directory and/or your "nodes" directory.

What to do if you experience delays

Sometimes, local conditions can affect the speed of the license manager. This section describes a few problems to look into if your license manager seems to be slow.

1. Is your DNS (or other name resolution) working well?

- DNS results are usually cached, but in some circumstances, a poor configuration or other problem in DNS can slow everything down.
- For the license manager, the main use of DNS is when finding the servers. If you are using the GHS_LMHOST variable, then the tool may need to look up the host you mention in it prior to connecting to it.
- You can use network addresses rather than host names to solve problems with slow DNS:
 - Suppose your GHS_LMHOST variable is set to: @license.ghs.com
 - Now suppose that the address for license.ghs.com is 10.2.1.4.
 - You can, instead, set GHS_LMHOST to 10.2.1.4 instead and avoid the DNS lookup every time a licensed tool starts up.

2. Are some hosts responding better than others?

Assume that you have two license servers. One is named lm1 and the other is named lm2. There are several cases to consider:

GHS_LMHOST=@lm1,lm2

- Both license servers will be contacted before you checkout a license.
- If one of these servers is unavailable, then there will be a delay while the tool times out contacting it.
- This is one of the most reliable settings. It can be slow when network conditions are bad, but it will be more likely to get a license if one is available.

GHS_LMHOST=lm1,lm2

- The tools will broadcast and try lm1 and lm2 explicitly.
- This will result in a few broadcast packets for every tool invocation. Depending on you network setup, this may or may not be acceptable.
- Using this method, the first machine to respond with available license will be selected, and the other machines will not be
- waited for.

GHS_LMHOST=@lm1

- The tools will connect directly to lm1.
- This is not as reliable as GHS_LMHOST=@lm1,lm2 because if lm1 is unreachable, you will not try lm2, but it will often be much faster.
- When too much broadcast traffic is a concern, this is often the best choice if occasional transient license failures are acceptable.

Other causes of problems

Here are a few other conditions to test for when diagnosing license manager problems:

1. Check NIS or other directory services.

- On UNIX machines, the Elan license manager uses the services database to determine which port it should run at. The license server uses `getservbyname ()` to look up the port for “ghs_elmd”.
- Most versions of unix have a file called `/etc/services`. This lists the name or port number of each known service. In most default configurations, this file is the only source searched for service information.
- Many systems/sites, however, use NIS, NIS+, or some other directory service to supplement or replace the `/etc/services` file. On Solaris, for example, `/etc/services` is frequently used during boot, but is replaced by an NIS map if NIS is started.
- The Elan license server expects that it will either get a port from the services database, or that the lookup will fail (in a reasonable amount of time.) After

such a failure, Elan will assume that it is to use its default port (udp port 8171.)

- If your system can take a long time to return a failure from `getservbyname`, then any of our tools trying to talk to the Elan server will have to wait for that failure prior to continuing.
- You can usually solve this problem by fixing your services database (NIS, NIS+, ... whatever) to return faster, or by adding `ghs_elmd` to the services database.
 - For many Unix systems, you could add the following line to `/etc/services`:

```
ghs_elmd      8171/udp
```

- If you are using a service like NIS, however, this line would have to be added to the source for the NIS map. Other directory services usually have a central location where a similar change could be made. Generally, such changes should be done by a network administrator since they will affect all machines in a domain.

2. "Server lm1 [ccom86]: Unable to write file"

- This error message is caused when a licensed tool or the license manager is unable to write to a file. The most likely cause of this is if the key directory (i.e. `/usr/green/keys`) is not writeable by the user who ran the license server.
- If the license manager is running as a privileged user, (root on UNIX, or Administrator on Windows), then usually it would have access to any file on the system. The main time when this is not true is when remote file systems are involved.
- If the license manager is having trouble accessing remote file systems, then you have three choices:
 - You can install the license server software and its key directories to a local disk.
 - You can get the remote administrator to change the export permissions on the remote file system so that the license server can access it. (On NFS, pay particular attention to the "root squash" option.
 - On Unix, you could run the license server under a different user id which does have access to the remote file system. This will probably require that you also change the ownership of the files currently installed in the

license manager directory. If the keys directory is /user/green/keys, and if the user who will run the license server is "USER", then one way to do this is to run the following commands(as a privileged user probably on the machine where the disks are physically located):

```
/user/green/keys
chown -R USER .
chmod u+w .
```

- If you make any changes to the license server setup, then be sure to restart the license server so that the changes can take effect.

If you get a "NetBIOS Error" while installing the licenses on a PC:

- This problem can be solved by regenerating your server code with the "30720/" prefix. This can be accomplished from the License Request Generator program in the Green Hills program group. After entering your information, and agreeing to the license agreement (if applicable), open the "Settings->Encoding" dialog box and enter 30720.
- Send this new server code to Green Hills with a description of the error so that we can help you resolve the problem.
- If you installed the rc script as /etc/init.d/ghs_elmd and linked it into /etc/rc2.d, or if you did something which is equivalent on your unix variant, then root will run the server on each boot.
 - Normally, root has permissions to write to any file on the system, but if your keys directory is located on a network directory, this may not be the case.
 - If the system or network administrator has setup the shared disks to "squash root" then any access from root is treated as if it had originated from the user "nobody".
 - To fix the problem, you need to do one of the following:
 1. Have the administrator change the share and/or mount options to allow root to access the directories.
 2. Chmod the directory so that it is world writable. This allows root to write to it even though it is treated as nobody.

3. Change your setup so that a user other than root runs the server. See below for information about non-root users.
 - If the license server is not being run as root, then the keys directory must have its permission set so that user can create and delete files in the directory. It doesn't matter if the write access is group or user access.
 - Here is one method of doing this when a non-root user named USER is to run the license server:
 - `cd /usr/green/keys`
 - `chown -R USER .`
 - `chmod u+w .`
 - Of course, these commands will have to be run by a privileged user (presumably root) and probably on the machine where the disks are physically located. If in doubt, check with a system administrator.
 - After making any of the above changes, make sure to restart the server under the correct user.

Index

Symbols

-? 14

A

about this manual
 conventions P-2
 fonts P-2
 keys P-3
aliases
 elmadmin 21

B

BROADCAST 13

C

client debugging 15
client ID
 listing 23
collection files
 key directory 19
command line 14
commands
 elmadmin 18
 elmver 32
 xelmadmin 23
communicating with the license server
 ghs_elmd 30
customizing Green Hills License Manager
 start-up time 30

D

debug output 15
debugging the client 15
debugging the license server 15
domain
 restricting A-2
dongle, hardware key license 10, 12

E

elmadmin 18
 aliases 21
 example

 installing a license key 21
 listing user and license information 23
 setting ready-key mode 22
hold period 23
hold time 19
key directory 19
options
 -b 18
 -C 19
 -c 18
 -d 19
 -e 19
 -h 19
 -i 19
 -k 20
 -l 20
 -p 20
 -q 20
 -r 20
 -s 20
 -v 20
 -x 20
 -z 21
shared licenses 19

elmalert 18
elmrpt 18
elmusage 18
elmver 18, 32
example
 installing a license key 21
 listing user and license information 23
 resource files A-5
 setting ready-key mode 22
excluded licenses
 resource files A-4

F

feature name A-4
 printing 20
floating license 10, 11
model 13

G

ghs_elmd 18

Index

- communication 30
- key directory
 - loading keys into 29
 - specifying 28
- key file 28
- using -n with 30
- zombie period 31

H

- held licenses A-2, A-4
- help command 14
- hold period
 - definition 19
 - listing 23
 - resource files A-4, A-5
- hostnames
 - BROADCAST 13
 - LOCAL 13

I

- installing a license 10

K

- key
 - directory
 - collection file 19
 - elmadmin 19
- key directory
 - elmd
 - loading keys into 29
 - specifying 28
- key files
 - elmd 28

L

- license administrator program 18
- licenses
 - considerations 11
 - dongle, hardware key 10
 - floating 10
 - installation 10
 - listing 20
 - node-locked 10

- requesting 10
- server
 - debugging 15

- LOCAL 13

N

- node-locked license 10, 12

O

- output retrieval 15

R

- ready-key
 - server code 22
 - setting ready-key mode 20, 22
- requesting a license 10
- reserved licenses
 - resource files A-4
- resource files
 - examples A-5
 - excluded licenses A-4
 - held licenses A-2, A-4
 - hold period A-4, A-5
 - key directory A-4
 - redundant servers A-2, A-3
 - reserved licenses A-4
 - restricting the domain A-2

S

- server code
 - xelmadmin 26
- shared licenses
 - elmadmin 19
 - LID 23
 - shared license count 19

T

- Troubleshooting 7

X

- xelmadmin

Index

GHS_LMHOST 24
server code 26
using 23

Z

zombie period 31