

*System Administrator's
Guide*



Copyright 2000, Rainbow Technologies, Inc.

All rights reserved.

<http://www.rainbow.com>

All attempts have been made to make the information in this document complete and accurate. Rainbow Technologies, Inc. is not responsible for any direct or indirect damages or loss of business resulting from inaccuracies or omissions. The specifications contained in this document are subject to change without notice.

Part Number 700619-001, Revision A

Software releases 7.1 and later

RAINBOW TECHNOLOGIES, INC.

50 Technology Drive, Irvine, CA 92618

Telephone: (949) 450-7300, (800) 852-8569

Fax: (949) 450-7450

RAINBOW TECHNOLOGIES LTD.

4 The Forum, Hanworth Lane, Chertsey, Surrey KT16 9JX, United Kingdom

Telephone: (44) 1932 579200

Fax: (44) 1932 570743

RAINBOW TECHNOLOGIES

122, Avenue Charles de Gaulle, 92522 Neuilly-sur-Seine Cedex, France

Telephone: (33) 1 41 43 29 02

Fax: (33) 1 46 24 76 91

RAINBOW TECHNOLOGIES GMBH

Lise Meitner Strasse 1, 85716 Unterschleissheim, Germany

Telephone: (49) 89 32 17 98 0

Fax: (49) 89 32 17 98 50

Additional offices are in the United States, Australia, China, India, the Netherlands, Russia and Taiwan. Distributors located worldwide.

SentinelLM is a trademark of Rainbow Technologies, Inc. Novell and NetWare are trademarks of Novell, Inc. Microsoft Windows, Microsoft Windows NT, Windows 95, Windows 98, and Windows 2000 are trademarks of Microsoft Corporation. UNIX is a registered trademark, exclusively licensed through X/Open Company, Ltd. All other product names referenced herein are trademarks or registered trademarks of their respective manufacturers.

Contents

Prefacexi
Note for Stand-alone Application Users	xi
Note for Networked Application Users	xi
About This Guide	xiii
Typographic Conventions	xiii
Syntax Conventions	xiv
Getting Help	xiv
Chapter 1 - Getting Started	1
Basic Concepts	1
What Is SentinelLM?	1
License Codes	2
Computer Fingerprints and Locking Codes	2
Stand-alone or Networked Licensing	3
Advantages of Network Licensing	4
Software Activation for Stand-alone Applications	4
How Do I Get a New or Upgraded License?	5
Getting Your Computer's Locking Code	6
Installing License Codes	7
Using Islic to Install a License Code	8
Software Activation for Network Licensing	8
Getting Your License Server Computer's Locking Code	9
Installing Network License Codes	9
Installing a Non-Redundant License Code	10
Installing a Redundant License Code	11

Using the SentinelLM Utilities	12
Chapter 2 - Configuring the License Server.	15
Overview	15
Additive and Exclusive License Codes	16
Location of the Applications and License Servers	17
Compatibility	17
License Server Structure	17
Performance Notes	18
License Authorization Process	18
Step 1: License Server Configuration	18
Step 2: Load License Codes	19
Step 3: Client Identifies License Server.	20
Step 4: Client Issues License Request.	20
Step 5: License Server Responds	21
Step 6: Client Renews License	21
Step 7: License Server Responds	21
Step 8: Client Returns License	21
Step 9: License Server Releases License	22
Configuring the License Server.	22
Setting the License File	23
Setting Usage Logging	23
Encrypting License Server Log File Entries	26
Setting Error Logging	27
Setting License Server Quiet Startup	28
Starting the License Server	28
User Privileges	28
Windows 95/98 User Privileges	28
Windows NT/2000 User Privileges	29
UNIX User Privileges	29
Starting the License Server as a Redundant License Server	29
System Requirements	30
Windows NT/2000 Service	30
Windows NT/2000 Event Logging.	31
Windows 95/98 License Server	31
NetWare	32

UNIX	32
Configuring the License Server Client.	32
Check the Network Protocol	32
Locate the License Server Computer.	33
Test the “Broadcast Method” of Locating a License Server	34
Test the “Named License Server Method” of Locating a License Server	34

Chapter 3 - Setting up Redundant License Servers and License Balancing . 37

Overview of License Balancing and Redundant License Servers	37
The Redundant License File (lservrlf)	40
The Redundant License Server Pool	40
The Redundant License Server Leader	40
Majority Rule	41
The Redundant License Free Pool	41
What Happens When a Redundant License Server Goes Down?	42
Overview of Setting up Redundant License Servers	42
Tips and Recommendations	43
Use Fixed IP Addresses	43
User Privileges Required	43
Using Redundant Commuter Licensing	44
Using Reservation Files	45
Enabling/Disabling Automatic License Balancing	45
Setting up the Redundant License Server Pool	46
Maintaining the Redundant License Server Pool	49
Adding a Redundant License to an Existing Server Pool	49
Redundant License Characteristics	50
Redundant License Server Pool Utilities	50
Using lslic to Add Redundant Licenses	51
Specifying Distribution Criteria for Redundant Licenses	53
Using wrlftool to Create or Edit a Redundant License File	54
Creating and Maintaining the Redundant License File	55
Adding/Editing/Deleting Redundant Licenses	57
Using rlftool to Create or Edit a Redundant License File	59
Using the rlftool Menu	59
rlftool Command-line Options	60
Using WlmAdmin to Maintain a Redundant License Server Pool.	61

What Does WlmAdmin Do?	62
Creating or Editing the Redundant License File	62
Creating or Editing the Reservation File	62
Monitoring License Servers and Licenses	63
Using lspool to Maintain a Redundant License Server Pool	63
Logging Events.	65
Sample Uses of lspool	66
To See How Tokens Are Distributed for a Particular License	66
To Distribute Tokens for a Particular License	67
Chapter 4 - Commuter Licensing	69
How Does Commuter Licensing Work?	69
Restricting Commuter Licensing	69
Commuter Licensing Authorization Checkin Policy	70
Locking the Authorization	70
Before Your User Checks an Authorization In or Out	70
Instructions for Your Network Users	71
Using Wcommute	71
Using lscmmute	74
Chapter 5 - Administrator Commands.	77
echoid - Get a Computer's Locking Code	79
ipxecho - Display the IPX Network Address	80
loadls - Install the License Server as an NT/2000 Service	81
lsdecode - Decrypt and Display a License Code	81
lserv, lserv9x, lservnt - Start the License Server	82
lslic - Install/Delete a License Code	84
lsmon - Monitor License Server Transactions	87
lsmon Display	88
lsrsvdown - Shut Down the License Server	89
lsusage - Display the SentinelLM Usage Log File	89
lsusage Display	91
Creating License Server Use Reports	93
To Import the CSV File	93
Creating Reports	94
lswhere - Display a List of License Servers	95

WlmAdmin - Administer License Management Activities	96
What Does WlmAdmin Do?	97
Monitoring Licenses and License Servers	97
Editing the Reservation File	98
Editing the Redundant License File	98
Shutting Down the License Server	98
Using WlmAdmin to Monitor Licenses and License Servers	98
Setting WlmAdmin Preferences	100
Selecting the License Servers	100
To Create a List of Defined License Servers	101
To Discover All License Servers on Your Network Subnet	101
To Refresh the Network Display	101
License Server-Specific Actions	101
To See Information on a License Server	101
To Add or Remove a License Server from the Defined Server List	102
To Shut Down a License Server	102
To Add a License Code to the License Server from the Keyboard	102
To Add a License Code to the License Server from a File	103
To Change the Log File for a Selected License Server	103
To Configure a Redundant License Server File	103
To Disable/Enable Token Borrowing for a License Server	103
License-Specific Actions	104
To See Information on a License	104
To Delete a License from the License Server	106
To Change the Token Distribution for a License	107
To Disable/Enable Token Borrowing for a License	107
Exiting WlmAdmin	108
Chapter 6 - Setting User Options	109
Setting User Alerts	109
Alert Specification	110
SCRIPT	113
EMAIL	113
Configuring E-mail Alerts for Windows	114
Configuring E-mail Alerts for UNIX	114
Setting Group Reservations	115

Including/Excluding Users or Computers	115
How the Reservation File Works	116
Reservation File Characteristics	116
Editing the Reservation File on Windows (WlsGrMgr)	117
Manually Editing the Reservation File	120
Appendix A - Using Environment Variables to Configure a License Server	123
Client - Commonly Used Variables	124
LSHOST - Set the License Server Name	125
LSFORCEHOST - Select a Single License Server	125
LSPROTOCOL - Set Communication Protocol	126
License Server - Commonly Used Variables	126
LSERVOPTS - Set License Server Options	126
License Server - Infrequently Used Variables	129
LSDEFAULTDIR - Set the Default Location of the License Server Files	129
LSERVRC - Set the License Code File	130
LSERVRCNF - Set the License Server Configuration File	130
LSRESERV - Set the License Server Reservation File	131
LSPORT - Set the License Server TCP/IP Port	131
Environment Variables for Administering License Queuing	132
LS_MAX_QWAIT_SEC - Set Maximum Queue Wait Time	132
LS_MAX_QHOLD_SEC - Set Maximum Hold Time	132
LS_MAX_QLEN - Set Maximum Queue Length	133
LS_MAX_GROUP_LEN - Set Maximum Group Queue Length	133
Notes on Setting Environment Variables by Operating System	134
Windows 95/98	134
Windows NT and Windows 2000	134
UNIX - Bourne or Korn Shell	135
UNIX - C or TCSH Shell	135
Stand-alone Application Use of Environment Variables	135
LSHOST - Disable a Network License	136
LSERVOPTS - Set Application License Management Options	136
LSDEFAULTDIR - Set the Default Location of License Management Files	136
LSERVRC - Set the License Code File	136
LSERVRCNF - Set the Application License Management Configuration File	137
Environment Variables Not Used in Stand-alone Mode	137

Index139

Preface

Welcome to the SentinelLM License Management system which protects the application you are using. SentinelLM may limit, track, or meter application use depending upon the unique requirements of your software license agreement.

Note for Stand-alone Application Users

If the SentinelLM-protected application you are using was protected for use only on a single computer (that is if your application vendor has not configured the application for network license management), then the only section in this book that applies to you is “Basic Concepts” on page 1 and “Software Activation for Stand-alone Applications” on page 4 in “Chapter 1 - Getting Started.” These sections contain an overview of license management and details on how to install a new license or upgrade an existing one. Please turn there now.

Note for Networked Application Users

The rest of the topics in this book apply to system and network administrators who are responsible for installing and maintaining SentinelLM-protected applications that make use of network license management. The sections that follow discuss how to:

- Install, configure, and start up network license servers.
- Set up and maintain a redundant license server pool.
- Use commuter licensing.

Note for Networked Application Users

- Configure SentinelLM with environment variables.
- Exclude or reserve licenses for users and groups of users.
- How to set up e-mail or script alert messages.

About This Guide

This guide gives all the steps for maintaining SentinelLM.

Chapter/Appendix	Description
<i>“Chapter 1 - Getting Started”</i>	Introduces license management and discusses how to obtain a new license from your vendor.
<i>“Chapter 2 - Configuring the License Server”</i>	Explains how to install and configure a SentinelLM license server.
<i>“Chapter 3 - Setting up Redundant License Servers and License Balancing”</i>	Gives an overview of redundant license servers and license balancing, discusses how to set them up, and contains information on utilities to maintain them.
<i>“Chapter 4 - Commuter Licensing”</i>	Provides instructions on using the commuter licensing utilities.
<i>“Chapter 5 - Administrator Commands”</i>	Contains instructions for using the SentinelLM administrator commands.
<i>“Chapter 6 - Setting User Options”</i>	Provides information on licensing options that can be configured after the protected application is installed.
<i>“Appendix A - Using Environment Variables to Configure a License Server”</i>	Explains how to use environment variables to configure a license server.

Typographic Conventions

The following typographic conventions are used throughout this guide:

Convention	Purpose
<i>italic</i>	Used to signal a new term, for placeholders, variables, and file names, or for emphasis.
bold	Used for command-line options and utility, dialog box, checkbox, menu, and command names.
<i>bold-italic</i>	Used for characters that you type, such as <i>logon</i> .
<code>courier</code>	This font denotes syntax, prompts, and code examples.

Syntax Conventions

The following syntax conventions are used throughout this guide:

Convention	Purpose
[]	Square brackets enclose optional syntax.
...	Ellipses indicate that a clause can be repeated.
	A pipe indicates that only one of the syntax choices it separates may be used.
{ }	Curly braces indicate that one of the options they enclose <i>must</i> be used in actual syntax.

Getting Help

For help with SentinelLM or with the protected applications that use it, please contact your vendor first. If additional assistance is needed, contact Rainbow Technologies at <http://www.rainbow.com>.

Chapter 1 - Getting Started

This chapter contains basic information of interest to anyone who uses or administers software that has been protected using SentinelLM. If your vendor has referred you to this manual, this is the chapter that you will want to read first.

Basic Concepts

This section contains information on basic concepts you will need to know about to administer licenses on your stand-alone or network computer.

What Is SentinelLM?

SentinelLM is a software toolkit that your vendor used to create the protected application you are using. The application uses SentinelLM to support the license agreement you have with your vendor. SentinelLM can authorize, meter, and report application use. Which of these capabilities is needed depends upon the type of application and details of your license agreement. When you start a protected application, the application first makes a check with SentinelLM to verify use is permitted by the license agreement. If the use is authorized, then SentinelLM gives the application permission to run. If permission is granted, this process is invisible. If permission is denied, then the application user will be informed and the application exits or the user is placed into a queue to wait for authorization. Permission may be denied because someone else is already using the application, the license has expired, or no license has been installed. If your license has expired or hasn't been installed, then you will want to continue reading this chapter. It tells how to obtain and install a license.

License Codes

SentinelLM captures your license rights in electronic form. These are stored as digital data called a *license code*. The license code indicates the maximum number of concurrent users of the application, on which computers it can be run, and the license expiration date, if there is one. License codes are stored in a file that is accessible to the SentinelLM software called the *license file*. Multiple license codes may be stored in a single license file.

The application may use one or more license codes. One license code may provide, for example, the right for four users to use the application at once, while another provides the right to two users to use the application at once. In this case, six users would be able to run the application at once using both license codes. If you need to add another concurrent user, this can easily be done by obtaining an additional license code from your vendor.

Note Throughout this book we talk about getting licenses and releasing licenses. Although it is convenient to talk about license management in this way, strictly speaking, licenses are never physically moved between the network license server and client computer. Instead, the license server simply keeps track of how many users can run the application and decrements and increments the license count as authorized users are granted permission to run the application and as they exit the application.

Computer Fingerprints and Locking Codes

Many applications are licensed to run on specific computers. Any restrictions on which computers may run the application are stored in the license code. When you purchase the application, if there are any computer restrictions, a license code must be created which includes the information needed to identify a specific computer. Information that can be used to identify a computer is called its *fingerprint*. This information might include the computer's network address or attached Computer ID key. Your vendor can select how much of this information is required to establish a computer's identity. The first step in creating a license code that is restricted to a particular computer is to run a utility that collects the information your vendor requires for computer identification. This utility col-

lects this information and outputs a *locking code* that is a number based upon the collected information. You provide this locking code to your vendor using a telephone or the Internet and your vendor produces a license code that specifies all authorized uses and includes the locking code.

Stand-alone or Networked Licensing

A protected application can be licensed to run on a single computer without using a computer network. SentinelLM calls a license for this type of application a *stand-alone license*. This kind of license is most often used when an application is used by a single person on a particular computer. The software is installed on that computer along with any needed license codes. This kind of application is often installed directly by the users themselves on their own computers.

As an alternative, a protected application can be licensed to run on multiple computers so that several computers can share it. In this case a computer network is required so that all computers that are authorized to run the software are connected. SentinelLM calls this a *network license*. A special SentinelLM program, called the SentinelLM *license server*, must be installed onto one or more computers in the network to coordinate the use of the application. The license server has access to the license codes. When the application starts, instead of communicating with SentinelLM software that has been linked to the application, the application communicates instead with the license server.

The same procedure for software authorization takes place in both the stand-alone and networked case. The only important difference is whether the SentinelLM license management software is part of the protected application or provided as a separate program, the license server.

Note If you have a stand-alone license, you may skip the rest of the information in this section. It simply describes some additional SentinelLM capabilities that are available with a network license. This information is of most value to a system or network administrator.

Advantages of Network Licensing

A network license adds many capabilities to license management. The following is a quick summary for the network administrator:

1. A network license server, by centralizing license use, allows you to keep track of how much an application is being used. (The license server usage log contains a record of each license transaction.)
2. You can restrict the subnet that a protected application can use to talk to a license server if needed.
3. You can use SentinelLM utilities to display a summary of current and historical software use.
4. You may impose local restrictions on software use. For example, a certain number of licenses may be reserved for particular departments or work groups.
5. You may balance the load of license use on your network and provide license server backup by creating a redundant license server pool.
6. You can configure the license server to report certain conditions such as approaching license usage limits.

Your application vendor may automatically install and configure a license server when you install a protected application. If this has been done, all you need to do is to obtain and install any license codes. If the license server wasn't automatically installed, then you will need to install it before installing the license codes. "Chapter 2 - Configuring the License Server" on page 15 tells how to install and configure the license server. The rest of this chapter tells how to obtain and install license codes.

Software Activation for Stand-alone Applications

A protected application won't run without a valid license code. Software is often licensed for a specific computer or network, but may be delivered and installed

without a license code. To obtain a license code, your vendor may require that you provide locking codes for all computers that it will run on. Your vendor will use this information to create the license codes that SentinelLM needs to authorize use of the protected application. Your vendor will then give you the license code along with instructions on how to install it.

Your vendor may also choose that a hardware key called a *Computer ID key* be used to identify a computer. In this case, your vendor can generate the license code that matches the Computer ID key and ship the hardware key and license code to you along with the software installation media. When you install the software, the license code can be installed automatically. This is a very convenient solution when you want to move the software from computer to computer.

Some applications are licensed without computer restrictions. Demonstration software is often licensed in this manner. Typically, demonstration software is licensed for a limited period of time so computer restrictions aren't required. This kind of software is often shipped with a license code that can be installed automatically. This software may then be used until the license expires. If you then try to run the application, you will be told the license has expired; to run the application again, you will need to contact your vendor to obtain a permanent license code.

Note If your application came with a license code, you can skip the next sections. It describes the procedure for getting a computer locking code and installing a license code.

How Do I Get a New or Upgraded License?

If your application didn't come with a license code or your previous license code has expired, then you will need to get and install a license code for that application. You may also wish to take advantage of additional features of the application or increase the number of authorized users. You can upgrade your software license by getting and installing a new license code.

In order to create a license code, the vendor may need some information about your computer. This is how the process works:

1. Contact your vendor following the instructions supplied. (Depending on your vendor, this may be by telephone or via the Internet.)
2. The vendor may tell you that a new license code has already been shipped with the application, and all you need to do is to install it. In this case, skip to the instructions in step #5, below.
3. If your vendor tells you a new license code is required, your vendor will probably ask you to get the locking code of your computer. You can find the instructions for doing this in “Getting Your Computer’s Locking Code” on page 6.
4. After you have the locking code, you must give it to the vendor. The vendor will create a license code using this information and provide you with a license code.
5. You will complete the license activation process by installing the license. If your application has been configured to accept the license code directly, the application will prompt you for the license code and install it for you. To find out if your application supports this, try running the application. If it asks for a license code, type it in and wait for the application to install the license. Once the application has installed the license, it is ready to run without any further attention. If the application does not ask for a license code, you must install the new license. See “Installing License Codes” on page 7 for instructions.

Once you have installed a valid license code, you are done.

Getting Your Computer’s Locking Code

Note Depending on how your vendor has asked you to install the license code, you may not need to get the computer’s locking code; the license installer may do it for you and display the locking code to you. However, it never hurts to use the **echoid** utility to get the locking code.

You use the SentinelLM **echoid** utility to get the computer's locking code. Your application vendor will provide you with instructions telling you what the locking code requirements and procedures are.

Sometimes your vendor will ask that you provide two different locking codes for a computer. This can be done to allow you to change the configuration of that computer sometime in the future.

Your vendor may have provided a Computer ID key. This key must be installed before you can produce a locking code.

From the operating system command prompt, type:

```
echoid
```

Write down the locking code or codes displayed by **echoid**. You can use **echoid** on a Windows 95/98, Windows NT/2000, or UNIX computer. For details on **echoid**, see "echoid - Get a Computer's Locking Code" on page 79 of "Chapter 5 - Administrator Commands."

Installing License Codes

Once you have been given a license code it must be installed where it can be found by the license management system. For a stand-alone application, the license code is installed in the same directory as the application. Your vendor may provide a program to install the license code. Or your vendor may have configured the protected application to install the license code. If neither of these is the case, then the **lslic** SentinelLM utility may be used to install the license code on a Windows 95/98, Windows NT/2000, or UNIX computer.

If your vendor has configured the application itself to install the license code, when you run the application and a valid license is not found, a license installation utility will guide you through obtaining a license code and getting a locking code. You may be presented with several different options for obtaining the license code, including using the Internet, calling your vendor on the telephone, or using a fax or e-mail service.

Using **lslic** to Install a License Code

If the application has not been configured to install the license code itself, you will need to use the **lslic** utility to install the license code you have obtained from your vendor.

Note Installing a stand-alone license is simple. If you don't want to use **lslic**, you can use a text editor to open the *lservrc* file; then, just type the license code into that file on its own line.

Your vendor should have placed the *lslic.exe* program in the same directory as your protected application. To install the license code, at the operating system command prompt, type:

```
lslic -A license-code
```

where *license-code* is the string of characters given to you by your vendor.

If you do not include the license code after the **-A** option, **lslic** asks you for it.

If your vendor has given you a license code *file* to install rather than a string of characters to type, at the operating system command prompt, type:

```
lslic -F filename
```

where *filename* is the name of the file given to you by your vendor containing the license code.

lslic adds the license code to the license file. The license file must reside in the same directory as the application, and by default is named *lservrc*.

Software Activation for Network Licensing

The main difference between activating software that uses a network license rather than a stand-alone license is that the license code must be installed in the license file that resides in the same directory and on the same computer where the license server runs.

You may also be asked to provide a locking code for the computer that will run the license server. (For even greater security, your vendor may also ask you for locking codes for the computers that will run the protected application.)

Getting Your License Server Computer's Locking Code

The procedure for getting a license server computer's locking code is the same as for a stand-alone computer (discussed above). Use the SentinelLM **echoid** utility to get the computer's locking code. Your application vendor will provide you with instructions telling you what the locking code requirements and procedures are.

Sometimes your vendor will ask that you provide two different locking codes for a computer. This can be done to allow you to change the configuration of that computer sometime in the future.

Your vendor may have provided a Computer ID key. This key must be installed before you can produce a locking code.

From the operating system command prompt, type:

```
echoid
```

Write down the locking code or codes displayed by **echoid**. You can use **echoid** on a Windows 95/98, Windows NT/2000, or UNIX computer. For details on **echoid**, see "echoid - Get a Computer's Locking Code" on page 79 of "Chapter 5 - Administrator Commands."

Installing Network License Codes

Once you have been given a license code it must be installed where it can be found by the license management system. For a network application, license codes are installed on the computer on which the license server runs. Your vendor may provide a program to install the license code. If not, then the **lslic** or **rlf-**

tool (or **Wrflftool**) SentinelLM utility may be used to install the license code on a Windows or UNIX computer.

Note If a network license code is intended for use on a redundant license server, it is called a *redundant license*. Redundant license codes are installed differently than other types of license codes. See below.

Installing a Non-Redundant License Code

Use the **lslic** utility to install the license code in the license file on the same computer and in the same directory in which the license server resides. By default, the license file is named *lservrc*, and it may contain multiple license codes. To install the license code, at the operating system command prompt type:

```
lslic -A license-code
```

where *license-code* is the string of characters given to you by your vendor.

Note If you need the licenses to be added to a remote host, set the LSFORCE-HOST environment variable on the computer on which you are using **lslic** to the host name of the computer containing the license server to which you want to add the license code.

If you do not include the license code after the **-A** option, **lslic** asks you for it.

If your vendor has given you a license code *file* to install rather than a string of characters to type, at the operating system command prompt, type:

```
lslic -F filename
```

where *filename* is the name of the file given to you by your vendor containing the license code.

Both of these commands install the license code permanently. **lslic** provides other ways to install a license code. For example, you can add the license code temporarily to be in effect only until the license servers are restarted. For details on **lslic**, see “lslic - Install/Delete a License Code” on page 84 of “Chapter 5 - Administrator Commands.”

Installing a Redundant License Code

Note In addition to providing license server backup, redundant license servers provide *license balancing*, which permits you to spread the load of license acquisition over multiple license servers. For example, suppose a license code allows 15 concurrent users of the protected application. (Each authorization to use the license is called a *token*. In this case, the license code contains 15 tokens.) You can distribute the license tokens among the license servers, perhaps allocating five tokens to each of three redundant license servers. Part of installing a redundant license is specifying the token distribution criteria.

There are several utilities you can use to install a redundant license (**lslic**, **rlftool**, and **Wrftool**). However, we recommend using **lslic** to add a redundant license because you will not have to stop and restart the redundant license servers for the change to take effect.

If you need the licenses to be added to a remote license server, set the LSFORCEHOST environment variable on the computer on which you are using **lslic** to the computer containing the redundant license server to which you want to add the license code even if you are running **lslic** on the license server's computer.

Note You may only use **lslic** to add redundant licenses after all the redundant license servers have been started with a redundant license file, *lservrlf*. For information on setting up a redundant license server pool, see “Chapter 3 - Setting up Redundant License Servers and License Balancing” on page 37.

To install the redundant license code, at the operating system command prompt type:

```
lslic -Ad license-code dist-crit
```

where *license-code* is the string of characters given to you by your vendor, and *dist-crit* is the distribution criteria of how you would like the number of users for that license distributed among the license servers. *dist-crit* takes the form:

```
server1:tokens1^server2:tokens2...
```

where *server* is the host name, IP address, or IPX address of the license server, and *tokens* is the number of license tokens you want to allocate to that server. Separate each server's allocation with a ^ symbol. If you do not include the license code after the **-Ad** option, **lslic** asks you for it.

If your vendor has given you a license code *file* to install rather than a string of characters to type, at the operating system command prompt, type:

```
lslic -Fd filename
```

where *filename* is the name of the file given to you by your vendor containing the license code.

Note If the file does not contain a distribution criteria, then license tokens will be distributed equally among all of the license servers. Or, to specify a distribution criteria, add the following line after the license code in the file:

```
#DIST_CRIT:server1:tokens1^server2:tokens2...
```

where the line *must* begin with the characters **#DIST_CRIT:**, and the *server:tokens* arguments follow the same format as discussed above.

Both of these commands install the license code permanently. **lslic** provides other ways to install a license code. For example, you can add the license code to the license server temporarily to be in effect only until the license server is restarted.

For more information on redundant license servers and on using **lslic** to add redundant licenses, see “Using lslic to Add Redundant Licenses” on page 51 of “Chapter 3 - Setting up Redundant License Servers and License Balancing.”

Using the SentinelLM Utilities

One or more SentinelLM utilities may have been provided with your protected application. These utilities help in the installation or configuration of the license management system, and most users will not need to use them. They are primarily used by network administrators to support network-licensed software. A few

of these utilities give information on license use or status and may be used by anyone. These utilities are described in “Chapter 5 - Administrator Commands” on page 77.

Chapter 2 - Configuring the License Server

This chapter describes how to start up and configure a license server. A license server is required when the protected application uses network license codes.

Note A *redundant license server pool* contains one or more license servers that are called *redundant license servers*. You can use redundant license servers to provide your network with license balancing and license server backup. For information on setting up a redundant license server pool, see “Chapter 3 - Setting up Redundant License Servers and License Balancing” on page 37.

Overview

The SentinelLM license server is a program that coordinates use of a protected application by multiple users and computers. This program usually runs on a computer that is located on the same subnet as any computers that need to run the application. The computers that run the application are called *clients*. When a protected application is started on the client, it sends a request for a license across the network to the license server. The license server grants the request if possible and returns an authorization message back to the client. Note that a license server program need not run on a file server system or any specific hardware server. Any computer that meets the requirements is acceptable.

Application use is authorized by license codes. Each license server is provided with a *license file* that contains one or more license codes. Each license code identifies the valid uses of a single application or feature. There may be multiple

license codes in a single license file. You generally add license codes to the license file by using the **lslic** utility.

Additive and Exclusive License Codes

License codes are either *additive* or *exclusive*. There can be multiple exclusive license codes per product in the license file. Only the last exclusive license code added to the file is used by the license server, if exclusive licenses are used. All other license codes, additive or exclusive, are ignored by the license server.

There can also be multiple additive license codes per product in the license file. Unless there are exclusive license codes for the same feature/version, then each additive license simply adds to the permitted uses defined by previous additive license codes. For example:

Let's say there is a product with a feature name *77* and version *5.0*.

There may be two additive license codes for *77 5.0*. One of these license codes may authorize five concurrent users. The other may authorize two concurrent users.

A license server provided with both license codes will authorize up to seven concurrent users of the application that uses feature name *77* and version *5.0*.

Additive license codes are generally used to increase the number of available licenses or number of client computers or to change the expiration date. Your vendor can tell you exactly what parameters have been changed by the additive license codes given to you.

Note Redundant license servers handle additive license codes slightly differently. A redundant license server may service both redundant and non-redundant license codes. A redundant license code that is additive will always override any exclusive or additive non-redundant license codes for the same application feature and version.

You may receive a permanent, non-redundant or redundant license code that replaces a temporary, trial license. Additive permanent licenses always replace trial licenses even if the trial license has not yet expired.

Location of the Applications and License Servers

The protected application is installed separately on each client computer. The license server can be installed and run on multiple computers. In this type of installation, the application running on a client may choose any available license server with a valid license code for that client.

It's possible to install the protected application and the license server on the same computer. In this case, the license server may authorize program use on that computer or on any other client computers for which there is a valid license code. The protected application can be authorized by the license server on the same computer or by a license server on any other computer. This makes it possible to locate the server on any computer, even a computer that is used to run one of the protected applications.

Compatibility

A license server running on a computer with one operating system can communicate with a client running on another operating system. It's also possible to have a client communicate with a license server running on one operating system today and a license server running on a different operating system tomorrow. This is all transparent to the user of a protected application. As long as the protected application and license server can communicate, all license requests will be correctly handled. There is also no difference between the license codes generated for license servers running on different operating systems.

In some cases, the application has been designed to work with either stand-alone or network license codes. In this case you can choose which method of operation you prefer by providing the appropriate license code.

License Server Structure

The SentinelLM license server is a System Service on Windows NT and Windows 2000. On Windows 95/98, the license server is a background program (although it optionally can be run in the foreground in a visible window). Under UNIX, the license server is a background process.

Performance Notes

High performance is always a consideration on a busy network. There are several things you can do to optimize the speed with which licenses are acquired by your network users, and to speed up the use of SentinelLM utilities:

- The computers on which license servers run should use static IP addresses rather than dynamically allocated IP addresses (DHCP). In fact, redundant license server computers *must* use static IP addresses.
- Client computers should be set up to directly contact the license server that services the licenses used by that client. The LSHOST and LSFORCEHOST environment variables or the *lshost* file can be used to do this.
- Client computers should contain a *hosts* file that maps IP or IPX addresses to the host names of the computers on which license servers reside; this speeds up contacting those license servers. The location of this file depends on your operating system. For Windows 95/98, this file is probably located in *\Windows*; for Windows NT/2000, this file is probably located in *\System32\Drivers\Etc*. The file itself contains instructions on entry format.
- Better performance can be achieved by running license servers on Windows NT/2000 rather than Windows 95/98.

License Authorization Process

The next sections give a step-by-step description of the entire license authorization process.

Step 1: License Server Configuration

When the license server is started it is configured in three stages. First, any options selected by command line parameters are set. Next, any options for

which SentinelLM has defined a specific environment variable are set. Finally, any options that are provided in the LSERVOPTS environment variable are set.

Many of the license server options can be set using all three methods. When a setting is made using more than one method, the later methods override any settings made by an earlier method. This flexibility helps in some situations, but in general it isn't needed. When we talk about setting an option in this guide, we recommend which method should be used to set the option.

This chapter discusses how to set the basic configuration options needed for most installations. The options needed to take advantage of a few of the advanced capabilities are discussed in "Appendix A - Using Environment Variables to Configure a License Server" on page 123. We also discuss license server command line options in "lserv, lserv9x, lservnt - Start the License Server" on page 82 of "Chapter 5 - Administrator Commands."

After configuration, the license server displays a startup banner unless the quiet startup option (**-q**) has been selected, in which case the license server starts without any messages.

Step 2: Load License Codes

After configuration, the license server reads the license file. The name and location of this file can be set using the license server options. The license server reads the file from start to finish, processing each license code in order. All additive license codes add to the licensing rights already processed. All exclusive codes override all previously read codes and any future additive codes.

If the license server can't find the license file, it will continue to run. When a license server is running it can also accept license codes across the network from other programs such as **lslic**. A server will process each received code as if it has just been read from the license file. The license server will also add the license code to the end of the license file if this was requested.

Step 3: Client Identifies License Server

If the application supports network licensing, when it is started it first tries to identify a license server. This is how it looks for a license server:

- If the LSFORCEHOST environment variable is set, the application looks for the specific license server host listed in that variable. If it cannot find that computer, an error message is displayed, and the application cannot run.
- If no LSFORCEHOST environment variable is set, the application looks for the LSHOST environment variable. If this variable has been set, then the application looks for any of the license server hosts listed. If LSHOST isn't set, then a check is made for a file with the name *LSHOST* or *lshost* in the directory where the application is located. If this file is found, then the application looks for any of the license server hosts listed in the first line of the file.
- If a license server isn't found using one of these methods, then a broadcast message will be sent out on the client computer's subnet requesting all license servers to identify themselves. The application will then choose one of these license servers. Using the broadcast method of finding the license server allows the server location to be changed without any actions on the part of the client computer. The broadcast is only used when the application starts, so it uses very little network bandwidth.

A protected application that has been built to support both stand-alone and network options can be converted to strictly stand-alone operation by setting LSHOST to NO_NET. (For instructions on setting environment variables, see "Appendix A - Using Environment Variables to Configure a License Server" on page 123.)

Step 4: Client Issues License Request

Once the application has identified the license server, it issues a request for one or more licenses to the license server. The server host name is translated to the

network address of the license server computer. The application sends a license request to this address and the license server's port.

Step 5: License Server Responds

When a license server receives a license request, it checks to see if a license can be granted based upon the current license codes. If it can, then an authorization is sent back to the client computer. Otherwise, the request is rejected and a denial is sent back to the client computer. In both cases, the activity may be logged to a file if this capability has been enabled.

Step 6: Client Renews License

Each license code specifies the maximum time period that a license may be held by an application without renewal. After a client has received a license, it must periodically request a license renewal from the license server. Forcing a license to be renewed allows a license server to reclaim a license when there is a problem with an application or a client computer. If the application doesn't renew its license within the allowed time, it loses its license and the server may grant it to another user. This action is handled by the application and is transparent to the user on the client computer.

Step 7: License Server Responds

The license server responds to a license renewal request in the same manner as the initial license request.

Step 8: Client Returns License

When an application ends, it sends a message back to the license server indicating that the license is no longer needed. This lets the license server immediately reclaim the license and make it available to another user.

Step 9: License Server Releases License

When the license server receives the license authorization back, it makes it available again to another user.

Configuring the License Server

If your application is being used with a network license then your vendor will provide you with a copy of the license server program and installation instructions. After you have installed the license server, you can then configure the license server options. If you are running the license server on several computers you need to complete this procedure separately for each computer.

If a copy of the license server has already been installed, perhaps for an earlier version of software or for a different application, then you only need to install a license server if your vendor provided you with a newer version. The way you determine the version number of the license server depends on the license server platform. For Windows 95/98, run the license server in foreground mode and select the **About** command on the **Help** menu. For UNIX computers, run the license server; it will display its version number. The newest SentinelLM license server can be downloaded from the Rainbow web page at <http://www.rainbow.com>.

The license server sets its configuration options only when it is first started. If your server is already running, you will have to stop it and restart it to make any configuration changes take effect.

You should review all of the configuration options that are described in this chapter during the initial license server configuration. In most cases, once these options are set they aren't changed. There are some advanced license server capabilities that can be enabled through other configuration options that are discussed in "Chapter 6 - Setting User Options" on page 109. These advanced capabilities can be enabled or disabled as needed.

Even though most license server configuration options can be set in several different ways, only the recommended way is described here. If you want to set

them in an alternate manner you should read “Appendix A - Using Environment Variables to Configure a License Server” on page 123 and “*lserv*, *lserv9x*, *lservnt* - Start the License Server” on page 82 of “Chapter 5 - Administrator Commands.” Please see those sections for a list of all the options available to you when you start up a license server.

Setting the License File

The license server must have read-access to a file that stores license codes. If you are going to use **lslic** or another program to install license codes, the license server must have read/write access to the license file when the license installation program is being executed. If you are having a problem installing license codes, then you should verify that the license server has read/write access to the license file.

Recommended settings:

- The default location of the license file is in the default license server directory. It’s best to leave the location of the license file there.
- For non-redundant license servers, the default name of the license file is *lservrc*. For redundant license servers, the default name of the license file is *lservr1f*. A redundant license server can service both redundant and non-redundant license codes if both types of license files are available to the license server.

Tip If the license server cannot find the license file and error logging is enabled, it will log the error in the error log file.

Setting Usage Logging

The license server can generate a usage log file. The name and location of the usage file is set by the license server startup options. The license server records all license requests and returns in this file. Usage reports can be generated using **lsusage**. (For information on **lsusage** and on creating reports, see “*lsusage* - Display the SentinelLM Usage Log File” on page 89 and “Creating License Server Use Reports” on page 93 of “Chapter 5 - Administrator Commands.”)

Information is recorded in the log file one entry per line in the following format:

Element	Description
Server-LFE	Customer-defined log file encryption level as specified by the license server -lfe option.
License-LFE	Vendor-defined log file encryption level. If this is non-zero, it overrides the Server-LFE.
Date	The date the entry was made, in the format: Day-of-week Month Day Time (hh:mm:ss) Year
Time-stamp	The time stamp of the entry.
Feature	Name of the feature.
Ver	Version of the feature.
Trans	The transaction type. 0 indicates an issue, 1 a denial, and 2 a release.
Numkeys	The number of licenses in use after the current request/release. (Encrypted if encryption level is set to 3 or 4.)
Keylife	How long, in seconds, the license was issued. Only applicable after a license release.
User	The user name of the application associated with the entry.
Host	The host name of the application associated with the entry.
LSver	The version of the SentinelLM license server.
Currency	The number of licenses handled during the transaction. (Encrypted if encryption level is set to 3 or 4.)
Comment	The text passed in by the protected application.

A typical entry might appear as:

```
# Startup SentinelLM v7.00 Fri Aug 13 17:33:31 1999 934590811 jsmith 140 4294489039
2 0 MTI1 Fri Aug 13 17:33:33 1999 934590813 99 v 0 1 0 jsmith jsmith 7.00 1 - - - - - MA== 754986
  OTM1MzQlNzk5
2 0 MTI1 Fri Aug 13 17:33:33 1999 934590813 99 v 2 0 0 jsmith jsmith 7.00 1 - - - - - MQ== 905242
  MTY2MDIyOA==
# Shutdown SentinelLM v7.00 Fri Aug 13 17:33:34 1999 934590814 jsmith 4294489039 Mg== 2058783
```

Here are what the fields in the first entry mean, reading from left to right:

Field	Description
2	(Server-LFE) Customer-defined log file encryption is set to level 2.
0	(License-LFE) 0 means the vendor set the log file encryption to customer-defined.
Fri Aug 13 17:33:33 1999	(Date) Date the license transaction was made.
934590813	(Time-stamp) Time-stamp of the transaction record.
99 v	(Feature/Ver) Feature name 99 and no version.
0	(Trans) License was granted.
1	(Numkeys) 1 token was in use after the request.
0	(Keylife) Only applicable for a license release.
jsmith	(User) The user requesting the license was jsmith.
jsmith	(Host) The host name of the computer on which the license request was made is also named jsmith.
7.00	(LSver) This log file is for a version 7.00 SentinelLM license server.
1	(Currency) Only 1 license was handled in this transaction.

SentinelLM writes records to the log file until it reaches its maximum size. The default value for maximum size is 1 megabyte but this can be changed using the **-z** option in the `LSERVOPTS` environment variable or at command-line. Once the maximum size is reached, the contents of the current log file will be saved into a backup file unless the no-backup **-x** option has been used in the `LSERVOPTS` environment variable or at command line. If **-x** has been selected, the license server will simply stop logging when the maximum size is reached. For information on the license server **-x** and **-z** options and for more detail on how log file backups are created, see “`LSERVOPTS` - Set License Server Options” on page 126 of “Appendix A - Using Environment Variables to Configure a License Server.”

Recommended settings (each of these can be set in the `LSERVOPTS` environment variable or by using the specified option on the command line):

- By default, usage logging is disabled. It’s best to enable usage logging by using the **-l** option. This log can be used to give you a very good idea of

how much each application is being used. If you don't have much disk space, you can restrict the size of the log.

You should enable logging and set the log file name and location. A typical log file name is *lserv.log*.

- It's best to use the **-z** option to specify the maximum size limit for the log file and to leave enabled automatic backup of the log file. You may want to set the maximum size of your log file fairly high to minimize the number of backup files created.

For a site with a 2 megabyte log file limit and no other license server options, **LSEROPTS** would just be set to **-z 2m**.

Tip As part of your payment arrangement with your vendor, your vendor may request that you do not use the **-x** option to stop log file backups.

Note If the SentinelLM log file contains characters that are not 7-bit ASCII (for example, if a user name contains multi-byte characters such as Japanese Kanji characters), those characters will not be viewable when looking directly at the log file. However, if you use the **lsusage -c** option to create CSV-format output from the log file, you will be able to view the multi-byte characters if you view the CSV-format file with an appropriate text editor on an operating system that displays the multi-byte language. You can also create Microsoft Access reports from the CSV-format file that can be viewed on a computer using the appropriate multi-byte operating system. (For information on creating reports from log file CSV-format output, see "Creating License Server Use Reports" on page 93 of "Chapter 5 - Administrator Commands.")

Encrypting License Server Log File Entries

For security reasons, the network administrator may wish to encrypt part or all of a license transaction record in the license server log file. To set the encryption

level, use the **-lfe** license server startup option and specify a number from 1 to 4, where:

Encryption Level	Description
1	No encryption.
2	No encryption. Transaction data will be readable, but tampering with or deleting an entry will be detected by lsusage . This is the default encryption level if you do not specify one (and if your vendor has not set an encryption level for your license codes).
3	Encrypt usage only. Transaction data will be readable except for license usage data. Such entries will not be displayed by lsusage .
4	Encrypt entire record. All transaction data for the license code will be encrypted. Such entries will not be displayed by lsusage .

Note For a specific license code, your application vendor can specify a particular encryption level; all entries for that license code will be encrypted to that level regardless of your encryption level settings. You may only set the encryption level of transaction records for license codes for which your vendor has allowed user-defined encryption.

For information on **lsusage**, see “lsusage - Display the SentinelLM Usage Log File” on page 89 of “Chapter 5 - Administrator Commands.”

Setting Error Logging

The license server can log unexpected conditions to a log file. The name of the error log file is specified by the license server **-f** startup option.

Normally error logging is disabled. If you are having a problem running or configuring the license server you may want to temporarily enable error logging. Then you will be able to capture all error messages in the log file.

Recommended setting: By default, error logging is disabled. We recommend you leave it disabled.

Setting License Server Quiet Startup

The license server normally displays a message upon program startup. This display can be suppressed if desired.

Recommended setting: By default, the startup message is enabled. We recommend you leave the message enabled.

Starting the License Server

License server installation and startup vary depending upon the operating system on the license server computer. The following instructions describe common server installation and start-up procedures. These instructions refer to license server command line options and environment variables. A detailed description of the command line options is provided in “lserv, lserv9x, lservnt - Start the License Server” on page 82 of “Chapter 5 - Administrator Commands.” The environment variables are described in “Appendix A - Using Environment Variables to Configure a License Server” on page 123.

User Privileges

Note that you must have the correct user privileges to configure or shut down the license server once it has been started.

Windows 95/98 User Privileges

The same user who starts the license server may also configure the license server (e.g., turn token borrowing on and off, change token distribution, turn logging on and off, and add or delete a redundant license server) and stop the license server.

Before allowing these actions, SentinelLM not only checks that the user name is the same, but also makes sure that the user is in the same network domain as the license server.

Windows NT/2000 User Privileges

Any user who has administrator privileges may start the license server and may also configure the license server (e.g., turn token borrowing on and off, change token distribution, turn logging on and off, and add or delete a redundant license server) regardless of who started it. Any user with administrator privileges can also use **lsrvdwn** to shut down the license server unless the license server was started by a user with the user name “Administrator,” in which case only “Administrator” can use **lsrvdwn** to shut it down.

Before allowing these actions, SentinelLM not only checks that the user has administrator privileges or is the “Administrator,” but also makes sure that the user is in the same network domain as the license server.

UNIX User Privileges

The same user who starts the license server may also configure the license server (e.g., turn token borrowing on and off, change token distribution, turn logging on and off, and add or delete a redundant license server). Only the same user who started the license server can stop the license server.

However, only a user with the user name “root” can use **lsrvdwn** to shut down the license server.

Before allowing these actions, SentinelLM not only checks that the user name is the same, but also makes sure that the user is in the same network domain as the license server.

Starting the License Server as a Redundant License Server

If a redundant license file (by default *lservrlf*) is in the same directory as the license server, when you start the license server it will start up as a redundant license server. If an *lservrlf* file is not in the same directory as the license server, you can still start it as a redundant license server if you use the license server **-rlf** option and specify the name and location of the redundant license file.

System Requirements

If you are using network licensing, a supported network protocol is required on the target machine.

Windows NT/2000 Service

The license server installation is done using the **loadls** utility. This utility may be run explicitly or as part of an installer program. This program is described in “loadls - Install the License Server as an NT/2000 Service” on page 81 of “Chapter 5 - Administrator Commands.”

Once loaded, the license server can be configured and started from the **Control Panel | Services** group. Automatic or manual license server startup can be selected.

If automatic startup is selected, the license server will be automatically started whenever Windows NT/2000 is started. The license server will remain running until Windows NT/2000 stops or the license server is stopped using the **Control Panel | Services** group.

If manual startup is selected, the license server must be started manually by using the **Control Panel | Services** group. When manually starting the license server, you may specify any license server command line options by using the **Startup Parameters** field in the **NT Services** window. This capability is not available when starting a service automatically. So, the recommended procedure is to select automatic startup and use the LSERVOPTS environment variable to configure the license server.

Note For Windows NT/2000, the LSERVOPTS environment variable only takes effect after you reboot the computer, and only if LSERVOPTS has been set as a System environment variable, not a user environment variable.

Administrator privileges are required to start and stop the license server service.

In **Service | Startup options**, use \\ in file paths instead of \. Or you can use /.

Windows NT/2000 Event Logging

For errors, look under **Applications** in the **NT Event Logger**. This log can be viewed using the Windows NT/2000 **Event Viewer** in the **Administrative Tools** group. After starting the **Event Viewer**, click the **Log** menu and select **Application** to view messages logged by the SentinelLM license server.

Windows 95/98 License Server

A Windows 95/98 license server can be configured to be started automatically when Windows 95/98 starts or to be started manually.

To enable automatic license server startup, add the license server program, *lserv9x.exe*, to the Windows 95/98 system **StartUp** folder. You may use **Windows Explorer** to open this folder, select the *lserv9x.exe* shortcut, and display its properties. You can review and change license server command line parameters by changing the contents of the **Target** field on the Shortcut tab.

To start the license server manually, you simply run *lserv9x.exe*. You can do this from a MS-DOS window or by double-clicking the *lserv9x.exe* entry in **Windows Explorer**.

By default, when a user logs out, the license server exits. This can be changed by providing the **-b** command line option to the license server. When this option is used, the license server will keep running as a background process when a user logs out.

When the license server is started without the **-b** option, a status window is created which is used as a reminder that the license server is running. If you see this Window or an entry for it in the taskbar, you will know that if you log out the license server will stop. You can also stop the license server by closing this window.

NetWare

IPX clients can be served by a Win32 server (*lserv9x.exe* or *lservnt.exe*) running on Windows 9x or Windows NT/2000 systems. This does not require an NLM or running any application on the NetWare server.

UNIX

The license server, *lserv*, is placed in an appropriate directory and appropriate file permissions are set. The execution permission must be set so that only authorized users may execute the license server.

The license server must be granted read permission to the license file and any other configuration files. The license server must also be granted read and write permission to any license serve log files that have been enabled.

The license server can be started automatically or manually.

To start the license server automatically when UNIX is started, the command line to execute the license server should be added to one of the operating systems startup files.

To start the license server manually, the command line can just be issued at any time.

Configuring the License Server Client

Protected applications which will be used with a license server and network license codes must be run on client computers that have been configured to communicate across the network with the license server.

Check the Network Protocol

The client computer and license server must support the same protocol.

To verify that TCP/IP is used on a Windows 95/98 or Windows NT 4.0/Windows 2000 computer:

1. From the **Control Panel**, double-click the **Network** icon.
2. On the **Configuration** tab (or **Protocols** tab on Windows NT/2000), look at the list of Network components installed to see if TCP/IP is listed.
3. Select the TCP/IP component and then bring up the **Properties** dialog box. Check the subnet mask. The client computer and any license servers it intends to use should be on the same subnet. If not, you must use the LSHOST or LSFORCEHOST environment variables to contact the proper license server.

Note If you are using DHCP (dynamic host configuration protocol), contact your network administrator for help in determining this information. Note that we recommend static, fixed IP addresses to give the best performance, and that redundant license server computers *must* have fixed IP addresses.

Locate the License Server Computer

The application program must be able to identify a license server. This can be done by having the application broadcast over the network for a license server or by providing the application with the name of the license server computer. In most cases, it's quicker to let the application use the broadcast method of finding a license server. With this method, you can change license servers at any time and no changes are required at the client computers.

Sometimes you may want to direct an application program to a specific license server. This may be to done to partition clients between license servers, or because the license server and client computer are located on different subnets or a large network and broadcast messages are undesirable.

Test the “Broadcast Method” of Locating a License Server

You can test the configuration using the following steps:

1. Verify the `LSHOST` and `LSFORCEHOST` environmental variables aren't set and there isn't an `LSHOST` or `lshost` file in the same directory as the protected application.
2. Run either the **lswhere** or **WlmAdmin** utilities on the client computer. These utilities do a network broadcast for license servers and display a list of all license servers that are visible from the client computer. **WlmAdmin** also displays a list of licenses; you can check that list to verify there is a server for the applications you want to run.
3. If all expected license servers are present, then the application will be able to run using the broadcast method. Of course, one of the license servers that is found must have a network license code that can be used to authorize use of the client computer.
4. If you are not able to see a particular license server, you should verify the network connection by using a network diagnostic program like **ping**. You can make this check by providing either the network address as a number (192.33.22.2) or as a name (`license_host`). If **ping** cannot find the server, then you have a network configuration problem that must be resolved before your SentinelLM-protected application can be used with a network license.

Test the “Named License Server Method” of Locating a License Server

You can test the configuration using the following steps:

1. Follow the steps given for the broadcast method until the client computer is able to find each license server that you want. If you are using the named license server method because the client computer and license server aren't on the same subnet, then you may want to skip right to the last step and just use **ping**.

2. If you can ping a license server when you specify its address by number but can't do this when you specify the address as a name, then your client computer has a problem with network domain name resolution. In this case you can either solve this problem or just provide the numeric form of the address to SentinelLM. Of course, if your license server is dynamically assigned a numeric network address, you will have to solve the name resolution problem. If you don't, then the next time the license server is assigned a new, numeric network address, the applications will no longer be able to find the license server.
3. Now set the LSHOST setting on the client computer to the name of the license server.

The LSHOST naming conventions are:

- Any valid name recognized by your network can be used.
 - Numeric names (192.22.11.33) can be used.
 - Symbolic names ("dept.computer") can be used.
 - You can provide a list of license server names. Each must be separated by a : symbol. The application program will try each license server in the list, beginning with the first listed, to find a license server that is serving this feature/version. The traversal will stop even if a license server is located that services this feature/version but it has no license tokens.
4. Test the LSHOST setting by running the **lsmon** utility on the client computer or specifying the computer on which the license server is running. If the name is correct, **lsmon** will be able to report status on the selected license server.

An application that supports both stand-alone and network license codes treats LSHOST as advisory. The application will first try to obtain a license by checking with all license servers listed in LSHOST.

5. You can now test the application. It should be able to obtain a license. If it doesn't then you should run **WlmAdmin** (for Windows) and verify

that the license server has a license that authorizes use of an application on the client computer.

Chapter 3 - Setting up Redundant License Servers and License Balancing

Redundant license servers and license balancing offer several important benefits to the network administrator:

- Your end users can acquire license tokens even if a particular license server goes down (license server backup).
- The speed with which an end user acquires a license token is optimized by distributing tokens among multiple license servers to reduce the traffic for a particular license server (license balancing).

The network administrator can initially allocate license tokens among multiple license servers, and SentinelLM will, over time, adjust the allocations to the optimal pattern based on actual license use.

This chapter gives an overview of redundant license servers and license balancing, discusses how to set up a redundant license server pool, and contains information on the utilities the network administrator can use to maintain the license server pool.

Overview of License Balancing and Redundant License Servers

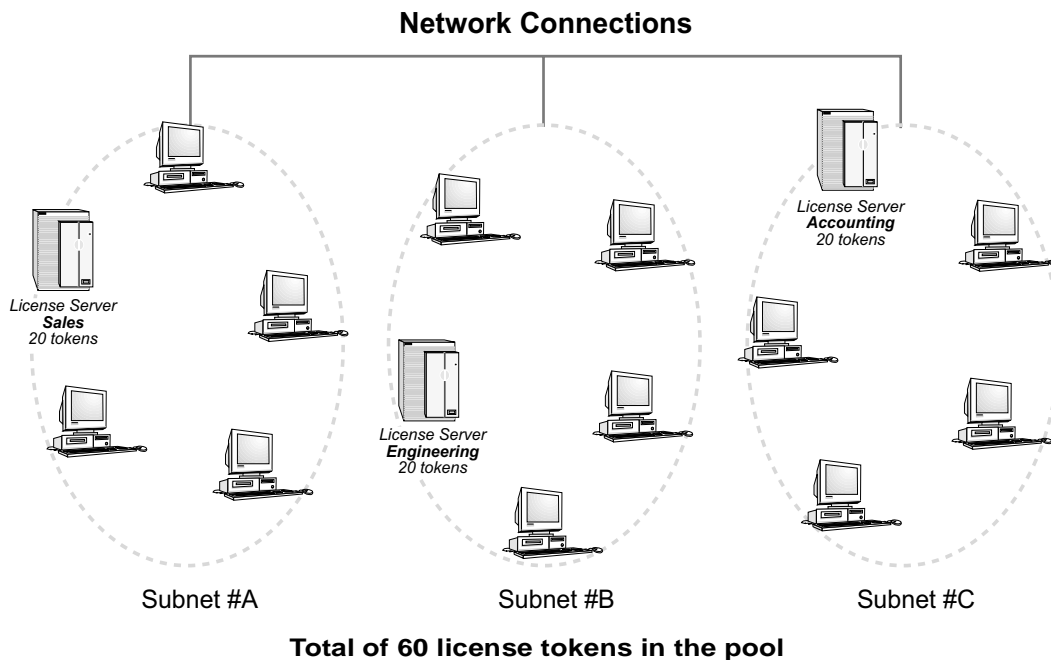
The method by which SentinelLM implements license balancing and license server backup is by using redundant license servers. All of the redundant license servers at your site form a license server *pool*. Each of the license servers in the

pool can take over for any other if one of the license servers goes down. Each license server runs on a separate computer on the network.

Imagine a wide-area network serving three basic groups of end users: the Sales Department, the Engineering Department, and the Accounting Department. Computers on this network might be in the same building, or might be in different cities or countries. You have a license code that permits 60 instances of a word processing application to run. (We say that the license code authorizes 60 licenses or that it has 60 *tokens*.)

The diagram below shows one possible configuration. You select three computers on the network and install a license server on each. Typically, each license server computer is on the same subnet as the majority of the end users that will acquire licenses from it. (However, each license server may be on a different subnet.) For example, the Sales Departments' computer is in the same building as the Sales employees, on the same part of the network used by those employees. This optimizes license management performance by speeding up license acquisition.

Figure 3-1: Sample Redundant License Server Pool



Note that the license server computers do not have to be on the same subnet—license servers can be in geographically separate locations on subnets connected to one another via WAN, Internet, or dial-up connections.

You decide to evenly distribute license tokens among all three license servers (20 each), assuming that each group of employees will use the protected application about the same number of times. This increases performance by distributing the load between multiple license servers.

However, if a group of employees ends up having a higher demand for licenses than the others, that group’s license server will borrow licenses from the other license servers automatically. This improves performance because users will be less likely to wait for a license from a distant license server.

To summarize: The configuration shown above uses redundant license servers to accomplish two goals: license server backup and license balancing.

You may also choose to configure the license servers so that only one is active and the others are solely backup license servers. In this case, all of the license tokens should be allocated to just a single license server and license balancing turned off. Then the other license servers would only be used if the main license server went down.

The Redundant License File (*lserverlf*)

The *redundant license file* (by default named *lserverlf*) is the key to the redundant license server pool. This file is created by the **rlftool** or **Wrftool** utility (or **WlmAdmin** calling **Wrftool**), and defines which license servers make up the redundant license server pool as well as how many tokens of which licenses are distributed among those license servers. Each computer on which a redundant license server resides will have a copy of this file.

Important Each license server in the redundant license server pool must be started with a copy of the redundant license file in the same directory in which the license server resides in order to come up as a redundant license server. (Or the license server **-lfe** startup option can be used to specify the location of the redundant license file.)

The Redundant License Server Pool

The redundant license file defines a group of redundant license servers called the *redundant license server pool*. The following sections give more information on the redundant license server pool.

The Redundant License Server Leader

All servers in a redundant license server pool can grant license tokens to clients. However, only one of the license servers is designated as the *leader* (the remainder of the license servers are known as the followers).

Tip Because the leader must communicate with all other redundant license servers, the leader should be in an area of the network with good bandwidth.

By default, the leader is the first license server in the pool you start up. However, **Wriftool** or **rlftool** can be used to set the priority order of the license servers. If the leader redundant license server goes down, the next highest-priority redundant license server becomes the leader in turn. To guarantee that a certain license server becomes the leader, use **Wriftool/rlftool** to give that license server a preference number of 1 and start it first before the other license servers.

The leader synchronizes communication between the license servers in the pool. The leader also makes sure that all the license servers in the pool are using the same version of the redundant license file. If this file is changed on one of the license server computers, after one of the license servers is stopped and restarted, the leader detects the change in the file and automatically updates the rest of the license server computers to the newer version.

If a license server exhausts its token allocation, it will ask the leader to borrow tokens from other license servers. If tokens are available from any other license server, the leader will take away the unused tokens and give them to the requesting license server. The auto-borrowing mechanism helps in load balancing also. Based on requests, the tokens get distributed among the license servers after some interval of time into a pattern that reflects real token use, leading to a balanced system.

Note Redistribution of tokens only occurs if license borrowing is turned on.

Majority Rule

Your vendor may require that your redundant license servers use *majority rule*. This dictates that a majority (over half) of the redundant license servers must be up and running for a request for that license to be serviced. For example, if you have defined five redundant license servers and majority rule is in effect, at least three of them must be up and running or a request for that license will fail.

The Redundant License Free Pool

In addition to distributing license tokens among redundant license servers, SentinelLM may also place some of those tokens in a *free pool*. (You can see how license tokens are distributed among redundant license servers and how

many are in the free pool by using **WlmAdmin** or **lspool**.) The free pool is a temporary storage location for license tokens. Tokens are placed in the free pool when you do not allocate all license tokens for a feature/version to the redundant license servers; leftover tokens go into the free pool.

As license balancing occurs, license tokens will be automatically redistributed from the free pool to the redundant license server requesting tokens. Or, the administrator can use **lspool** to clear the free pool by increasing the number of tokens distributed among the redundant license servers. When the free pool is empty and a license server requests more tokens, the leader license server takes away unused tokens from other license servers and assigns them to the requesting license server.

What Happens When a Redundant License Server Goes Down?

If a redundant license server goes down, the license tokens it was allocated are inherited by the other redundant license servers, and the end user now acquires license tokens from another license server with no administrator intervention necessary.

If majority rule is in effect for a particular license code (see “Majority Rule” on page 41) and the majority of redundant license servers go down, requests for licenses for that feature/version are no longer serviced until the majority of license servers are up and running again. (Requests for non-redundant licenses or redundant license codes not using majority rule will continue to be serviced.)

Overview of Setting up Redundant License Servers

There are three aspects to setting up and using redundant license servers. This section is an overview of this process. Following sections give more information.

1. First, you must decide how many redundant license servers to set up and select the computers on which they will reside. (You must define at least three; we recommend a maximum of five.) Various factors, including

network performance, affect this decision. If you want the license servers to be locked to specific computers, you must give your vendor the locking code for each computer on which a redundant license server will be installed.

Your vendor includes the computer locking code when defining the license code.

2. You create the redundant license file, *lservrf*, using the **rlftool** or **Wriftool** utility (or by using **WlmAdmin** to call **Wriftool**) to define the redundant license server pool.
3. Then you bring up the redundant license servers. To maintain the pool, use the **lspool** and **WlmAdmin** utilities to dynamically reconfigure the redundant license server pool and set license token allocation.

Tips and Recommendations

The following sections contain information you should be aware of before setting up the redundant license server pool.

Use Fixed IP Addresses

Each computer on which a redundant license server resides must have a fixed IPX address. Therefore, do not use DHCP to dynamically allocate IP addresses for those computers.

Using fixed IP addresses for client computers will also increase SentinelLM performance.

User Privileges Required

Setting up redundant servers requires you have access privileges to all computers and areas of the network on which license servers will be installed. If you make changes to the redundant license file, those changes will not be distributed to all license servers in the redundant license server pool unless you have network and write access to each of the redundant license server computers.

When using the SentinelLM utilities, you must have administrator privileges on Windows NT or Windows 2000, be logged into the root on UNIX, or be the user who started the license server on Windows 95/98 to:

- Change distribution criteria.
- Shut down the license server (and on Windows NT/2000, if user name Administrator started the license server, only user name Administrator can shut it down using **lsrvdown**).
- Turn license server logging on and off.
- Turn token borrowing on and off.
- Add or delete license servers from the redundant license server pool.

For more details on user privileges, see “User Privileges” on page 28 of “Chapter 2 - Configuring the License Server.”

Using Redundant Commuter Licensing

A commuter license enables your end user to “check out” an authorization for a license for temporary use on a portable computer. (See “Chapter 4 - Commuter Licensing” on page 69 for general information on using commuter licenses.) Your vendor selects whether a network license code will be a commuter license when creating the license code. Commuter licenses can also be redundant licenses; if they are, their tokens can be distributed among the redundant license servers just as with other redundant license codes.

Your end users can use **Wcommute** (for Windows) or **lcommute** (Windows or UNIX) to check out a license authorization from any of the redundant license servers. However, the customer must check that authorization back in to the same license server from which it was checked out. Because commuter licenses will automatically expire in 30 days or fewer after they are checked out, there is no real reason for your customer to check the authorization back in unless the redundant license server pool is running low on commuter license tokens (which you can check by using **lsmon** or **WlmAdmin**).

Using Reservation Files

You may use reservation files to reserve license tokens for particular users or groups of users. The reservation file applies to a specific license server, and may not reserve more than the total number of tokens initially allocated to that license server.

Members of a single group can request as many tokens as have been reserved for them, even if the license server doesn't currently have that many; if borrowing is enabled, additional tokens will be borrowed from other license servers. Members outside of that group cannot use tokens reserved for the group.

Only tokens not reserved for a group can be borrowed by other license servers. For example, if a license server's allocation is 10 tokens and all of them are reserved, this license server will not grant tokens to another license server requesting tokens.

For details on reservation files and on using **WlsGrMgr** to create and edit the reservation file, see "Chapter 6 - Setting User Options" on page 109.

Enabling/Disabling Automatic License Balancing

By default, automatic license balancing is enabled for redundant license servers. The way license balancing is achieved is that when one redundant license server is running low on license tokens, it *borrow*s tokens from another redundant license server. Because SentinelLM automatically determines the optimal license balance among the redundant license servers, you do not need to do anything else to balance license tokens.

If you want to use license balancing, at least one license token must be allocated to each redundant license server. You may choose to turn license balancing off by disabling token borrowing or by assigning all tokens to a single redundant license server (zeroing the rest of the license servers).

If borrowing is turned off for all license servers, only one license server will be able to issue license tokens. All clients must connect to that license server. If that license server fails, another license server automatically takes over and all clients are automatically reconnected to the new license server. **WlmAdmin** or **lspool**

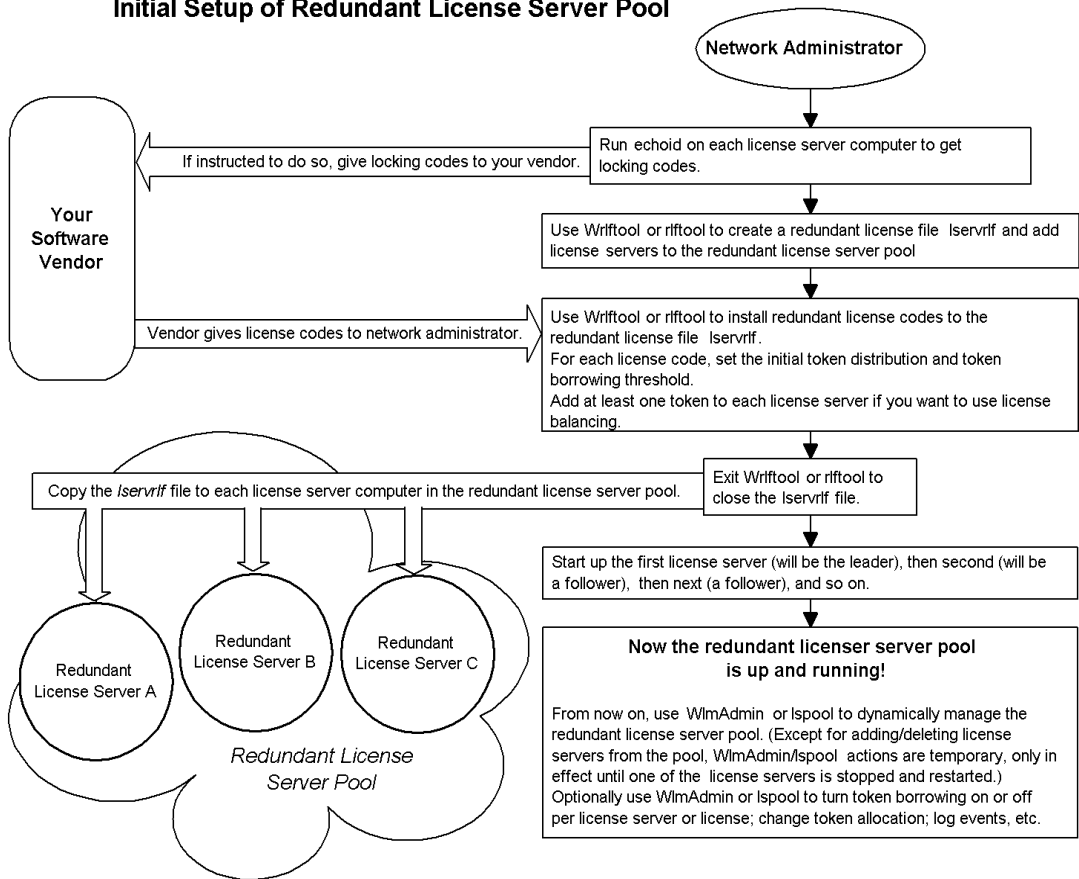
are used to turn borrowing off on a specific license server for a particular license code feature/version or all feature/versions.

Wrlftool (for Windows) or **rlftool** (for Windows or UNIX) is used to define the *borrowing threshold*, the percentage of license tokens that, when consumed, will trigger borrowing from another license server in the pool. For example, if a license has 100 tokens (that is 100 concurrent users can use the license) and the borrowing threshold is set to 90%, then when 10 license tokens are remaining on any license server, that license server will borrow more tokens from another license server in the pool. By changing the threshold, you can adjust the amount of token borrowing that occurs to fine-tune network performance.

Setting up the Redundant License Server Pool

The following diagram gives an overview of how to initially set up a redundant license server pool. For more information and for details on using the utilities referenced, see the next section.

Initial Setup of Redundant License Server Pool



1. First, install (but don't run) the license servers on different computers on the network. Note that these license servers must be the same version, at least SentinelLM version 7.0.
2. Next, set each client to access the preferred license server for that client by setting the LSHOST or LSFORCEHOST environment variables on that client's computer. (Or, if preferred, you can choose not to set this

environment variable and allow the client to broadcast for an available license server.)

3. If your vendor has specified that the license servers will be locked to specific computers, you will need to run **echoid** on each computer and record the locking code. You should also record the host name of each computer and its IP or IPX address. You will then give the locking codes, host names, and addresses to your vendor so that he or she can create the license codes for these license servers. For information on using **echoid**, see “echoid - Get a Computer’s Locking Code” on page 79 of “Chapter 5 - Administrator Commands.”
4. You will now install the protected applications on the end user computers.
5. If any license servers are running that will be used as redundant license servers, you must shut them down before creating the redundant license file.
6. After you receive the license codes from your vendor that will be serviced by the redundant license server pool, use **Wrlftool** or **rlftool** (or **WlmAdmin** to call **Wrlftool**) to create a redundant license file to:
 - Add the license servers to the redundant license server pool by specifying the host name and IP or IPX address for each computer containing a redundant license server. This defines which license servers are in the redundant license server pool. Note that all license servers in the pool must use either IP *or* IPX, but you cannot mix the protocols.
 - Set the preference number of the license servers. The first license server added to the pool and started up first is also known as the leader and all other license servers will be known as the followers. The preference order specifies the order in which the leader will be chosen if license servers go down.
7. Use **rlftool**, **Wrlftool**, or **WlmAdmin** to add the redundant license codes received from the vendor into the redundant license file and set their initial token distribution among the redundant license servers.

8. Since this is the first time that the redundant license server pool has been set up, you need to copy the redundant license file to each of the computers on which the redundant license servers resides. (After this, changes to any one of the redundant license files will automatically be transmitted to each redundant license server when one of the license servers is stopped and restarted.)
9. Bring up each of the license servers in the pool. Because the redundant license file is in the same directory as each license server, each license server will automatically start up as a redundant license server.
10. You can now use **WlmAdmin** or **lspool** to turn off token borrowing (license balancing) entirely, or enable/disable it for a particular license feature/version.

Maintaining the Redundant License Server Pool

Once redundant license servers are set up, you can use **lspool** or **WlmAdmin** from any computer on the network to change the license distribution, view information about the redundant license server pool, and turn borrowing on/off. You must set the LHOST environment variable to point to one of the redundant license servers when using **lspool**. Some **lspool** options dynamically change the redundant license server configuration, but do not write the changes permanently to the redundant license file. When the redundant license servers are restarted, the changes are lost. Other **lspool** options make permanent changes.

Information on **WlmAdmin** and **lspool** is included below.

Adding a Redundant License to an Existing Server Pool

By using the **lslic** utility, redundant licenses can be added dynamically. If the requested server is the follower, it will forward the request to the leader; the leader server will then check the validity of license. If the license can be added, the leader will add that license and forward the license code to all the followers.

If distribution criterion is not specified using **lslic**, the tokens will be distributed equally among the redundant license servers.

Redundant License Characteristics

A redundant license code may be additive or exclusive. If it is exclusive, it acts like any other exclusive license code. If it is additive, when it is installed it:

- Overwrites any exclusive or additive non-redundant license code for the same feature/version.
- Combines any redundant additive license codes for the same feature/version unless aspects of the older additive license codes differ in important respects.

Redundant License Server Pool Utilities

A number of utilities can be used to create and maintain redundant license server pools. The table below lists the most commonly used functions of these tools to give you a quick reference summary of which tool to use for what.

See the descriptions of these tools later in this chapter for a more complete list of functions. Note that **Wrlftool** and **WlmAdmin** can be used on Windows computers only, while the other utilities may be used on Windows and UNIX computers.

Table 3-1: Redundant License Server Pool Utilities Summary (T = Temporary change in effect only until license server is stopped and restarted; P = Permanent change to redundant license file, I = Informational only, makes no changes)

Function	Utilities		
	Islic	WlmAdmin/ Ispool	Wriftool/ rlftool
Create redundant license file	--	P ^a	P
Add/delete license servers to/from existing pool	--	P ^a	P
Add/delete license to/from pool and set initial distribution of tokens	T/P ^b	P ^a	P
Set borrowing threshold	--	P ^a	P
Change distribution of tokens for an existing license	--	T, P ^a	P ^c
Turn token borrowing on/off	--	T	--
Display current, dynamic, token distribution	--	I	--
Set/change pool name	--	P ^a	P
Display list of license servers in pool	--	I	--
Set license server preference order	--	P ^a	P

- a. When done from within **WlmAdmin** by using the **Edit | Redundant License File** command, which calls **Wriftool**.
- b. **Islic** provides options to add a license code permanently to the redundant license file or add a license code to the license servers' memory (to be in effect only until all redundant license servers are stopped and restarted again).
- c. Although **Wriftool** and **rlftool** write the license token distribution to the redundant license file, this change cannot be considered permanent because, using the token distribution in the redundant license file as a base, SentinelLM continues to adjust token distribution to the optimum pattern based on real license use. You may turn borrowing off after license server startup to make the distribution permanent.

Using Islic to Add Redundant Licenses

The **Islic** utility can be used to add non-redundant license codes to the license file (*lserverc*) or redundant license codes to the redundant license file (*lserverlf*), depending on which options are used. (For information on using **Islic** for non-redundant license codes, see “Islic - Install/Delete a License Code” on page 84 of “Chapter 5 - Administrator Commands.”)

The **lslic** command installs or deletes a license. If you do not include necessary information on the command line, **lslic** will ask for the missing information. When you enter a license code, if you are entering an encrypted license code, type only the characters before the # symbol.

lslic can add or delete licenses without stopping and restarting the license server, but the change is temporary unless you use an option that makes the change to the license file.

To add redundant license codes:

1. Set the LSFORCEHOST environment variable to point to the remote license server containing the redundant license file to which you want to add the license codes. (Or use **lslic** on the computer on which the license server resides in the same directory as the license server.)
2. On the operating system command line, type:

```
lslic option
```

where *option* is one of the following **lslic** options:

Option	Description
-ad <i>license-code dist-crit</i>	Adds a license code to the redundant license servers without updating the redundant license file. This change is temporary and in effect only until the license server is restarted. If either the license code or the distribution criteria (how the tokens of that license code will be distributed) is not specified, lslic will prompt for that information. See below for information on the <i>dist-crit</i> format.
-Ad <i>license-code dist-crit</i>	Adds a license code to the redundant license servers and updates the redundant license file. This change is permanent. If either the license code or the distribution criteria (how the tokens of that license code will be distributed) is not specified, lslic will prompt for that information. See below for information on the <i>dist-crit</i> format.
-fd <i>filename</i>	Adds all the license codes and distribution criteria in the file to the redundant license server's memory. (Change is temporary until the license server is restarted.) See below for information on the file format.

Option	Description
-Fd <i>filename</i>	Adds all license codes and distribution criteria in the file to the redundant license server and updates the redundant license file. (Change is permanent.) See below for information on the file format.
-d <i>feature version</i>	Deletes a feature/version from the contacted license server. (Change is temporary until the license server is restarted.)
-removeall	Removes all license codes from the license server but does not delete them from the license file. (Temporary change.)

Specifying Distribution Criteria for Redundant Licenses

When using the **-ad** and **-Ad** options, the format for distribution criteria is:

```
Server1:tokens1^Server2:tokens2...
```

where *Server* is an IP address, IPX address, or host name of the computer containing a redundant license server and *tokens* is the number of license tokens to be allocated to that redundant license server. Note that all *server:tokens* arguments must be separate with a ^ symbol. For example, to distribute three license tokens to each of three redundant license servers:

```
Sales:3^Engr1:3^Accounts:3
```

When using the **-fd** and **-Fd** options, the format of the file is the following, with each element on a separate line in the file:

```
license code 1
#DIST_CRIT: server1:tokens1^server2:tokens...
license code 2
#DIST_CRIT: server1:tokens1^server2:tokens...
```

Note If a license code is not followed by a **#DIST_CRIT:** line, the license tokens are evenly distributed among the license servers in the pool.

The distribution criteria line must begin with the characters **#DIST_CRIT:**. The format of the data following the **#DIST_CRIT:** label is the same as for the **-ad** and **-Ad** options. For example:

```
#DIST_CRIT:Sales:5^192.160.2.101:5^Engr1:5
```

Using Wrlftool to Create or Edit a Redundant License File

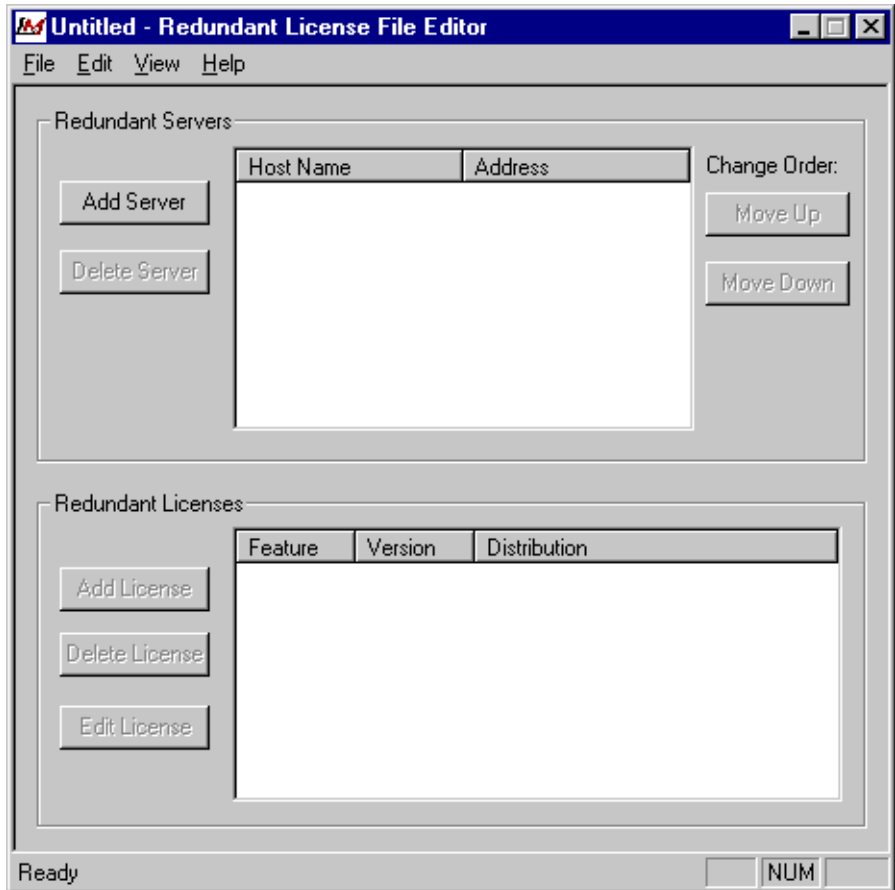
Wrlftool is a Windows-interface program that allows you to create and maintain a redundant license file.

Note If you are editing an existing redundant license file, to avoid synchronization problems because someone else is adding licenses or changing distribution criteria in the redundant license file while you have it open, we recommend you make a copy of the file under another name, make your changes, and then rename it back to its original name when you are done.

The changes you make to the redundant license file will not take effect until you copy the changed file to a redundant license server in the pool (or edit it directly on the license server using the precautions discussed above) and then restart that server.

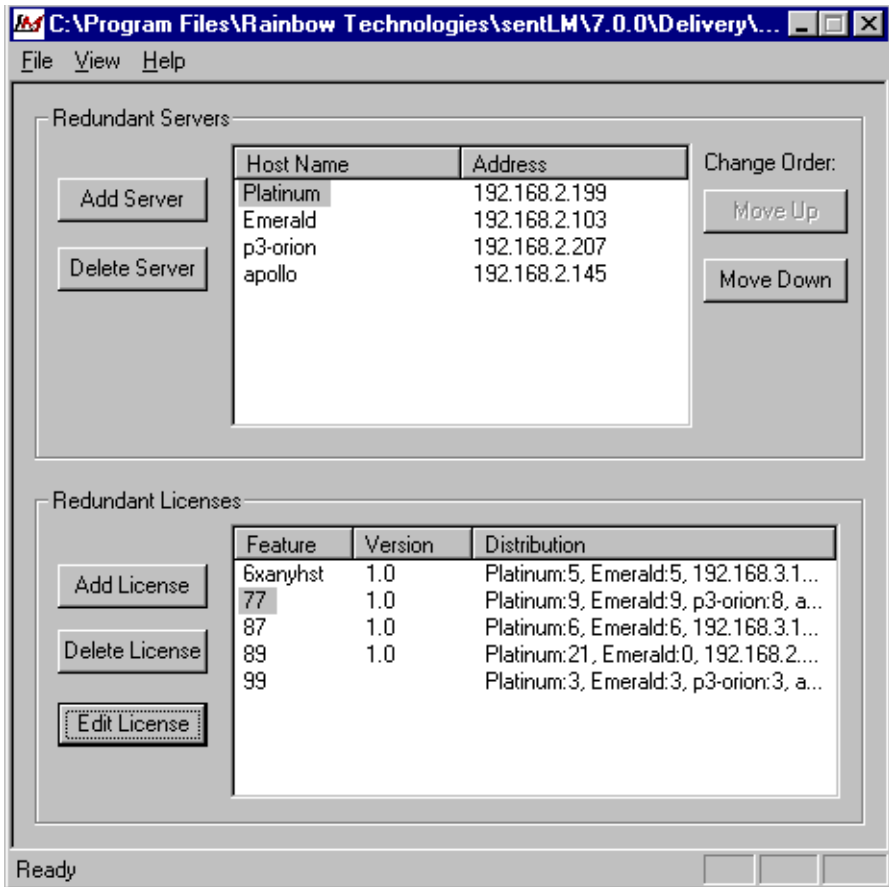
1. Using Windows Explorer, navigate to the location of the *wrlftool.exe* file, and double-click the file to start the program. (The exact location of this file depends on your application vendor.)

You now see the main **Wrlftool** display.



Creating and Maintaining the Redundant License File

1. To create a new redundant license file, go to the **File** menu and select **New**. To edit an existing redundant license file, go to the **File** menu and select **Open** and then select the file.



2. To add a new license server to the pool, click **Add Server**. You are then asked to type the host name and the IP or IPX address of the computer on which that license server resides. All license servers in the pool must use either IP *or* IPX to communicate, but you cannot mix the protocols.
3. Once you have added more than one license server to the pool, you can select one of the license servers and click **Move Up** or **Move Down** to adjust its order in the pool. This sets its preference order. Although the

first license server in the pool to be started up becomes the leader by default, the preference order determines in what order the license servers will become the leader if license servers go down.

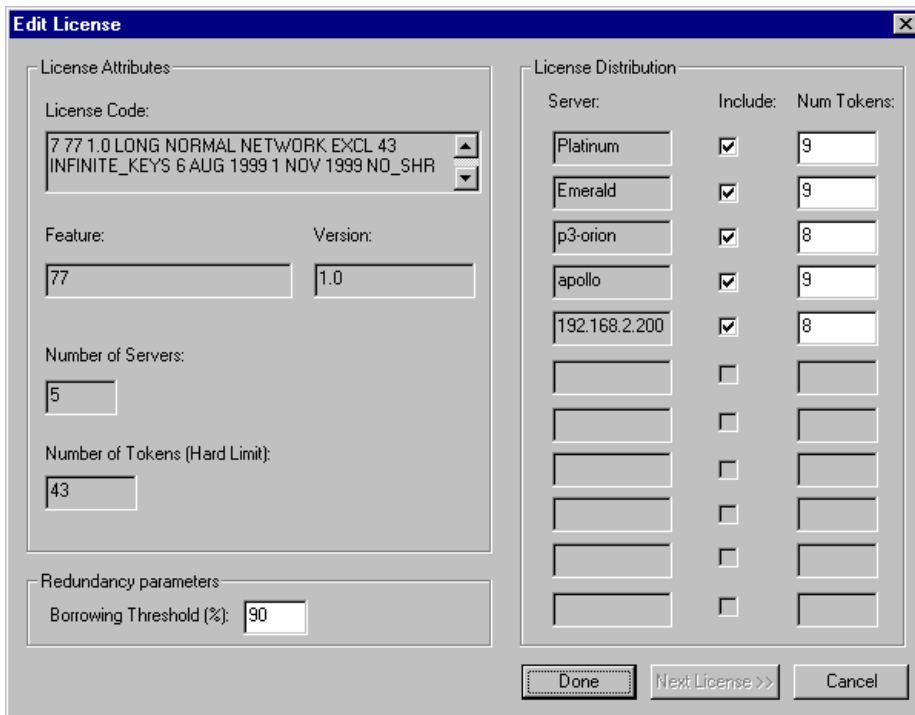
4. At any time you can select a license server and click **Delete Server** to delete it from the pool.

Adding/Editing/Deleting Redundant Licenses

1. To add a redundant license to the pool, click **Add License**.

You see a screen that allows you to either add the license code or specify a file containing the license code.

2. Once you have added the license code, you see the license editing screen:



- The **Edit License** screen displays information about this license such as the feature/version licensed, how many license servers it is set up to use, and so on. This screen allows you to set the following information for this license:
 - Borrowing threshold** - This is the percentage of license tokens that, when consumed, will trigger borrowing from another license server in the pool. For example, if this license has 100 tokens (that is 100 concurrent users can use this license) and the borrowing threshold is set to 90%, then when 10 license tokens are remaining on any license server, that license server will borrow more tokens from another license server in the pool.

- **License distribution** - You see a list of all license servers in the pool. You can choose which license servers will service this license by selecting the appropriate **Include** check boxes. For each license server you include, you can select how many tokens will initially be distributed to that license server.
4. After at least one license has been added to the pool, you can select an existing license from the **Wrftool** main screen and click **Edit License** to change its token distribution and borrowing threshold as explained above. You can also select a license and click **Delete License** to delete it from the pool.

Using rlftool to Create or Edit a Redundant License File

rlftool is a command-utility for Windows and UNIX. (On a Windows computer, use **rlftool** from within an MS-DOS window.) It may also be used in menu mode on Windows computers from an MS-DOS window.

Note If you are editing an existing redundant license file, to avoid synchronization problems because someone else is adding licenses or changing distribution criteria in the redundant license file while you have it open, we recommend you make a copy of the file under another name, make your changes, and then rename it back to its original name when you are done.

The changes you make to the redundant license file will not take effect until you copy the changed file to a redundant license server in the pool (or edit it directly on the license server using the precautions discussed above) and then restart that server.

Tip If you are running on a Windows computer, you may find using the Windows-interface **Wrftool** program more convenient than using **rlftool**.

Using the rlftool Menu

To use **rlftool** in menu mode to create a new redundant license file, from DOS type:

```
rlftool
```

Or, to modify an existing redundant license file, type:

```
rlftool -l redundant-license-file
```

You then see a menu that lists the **rlftool** options. Type the number of the option you want to use and press Return. (At a minimum, you will want to use option 1 to create a new file, option 2 to add license servers to the pool, and then option 7 to add license codes.) To exit and save the file, type **12** and press Return.

For information on the **rlftool** options, see the descriptions in the next section. Two options available when using **rlftool** in menu mode that are not available as command-line options are:

- **Preference Order** - The first redundant license server in the pool to be started is by default the leader. However, by setting the preference order, you set the order in which the license servers will be elected the leader if the leader goes down.
- **View/Edit License** - You can select an existing license to view, and may change token allocations for that license.

rlftool Command-line Options

To use **rlftool** as a command-line utility, type **rlftool** followed by any of the options below except **-l** (which will take you to the menu). The **rlftool** options are:

Option	Description
-h	(Help.) Displays list of rlftool options.
-l filename	Load or create a redundant license file.
-a server-name address	Add the license server specified by the host name and IP address/IPX address given to the redundant license server pool.
-d server-name	Delete the license server specified by the host name from the redundant license server pool.

Option	Description
-A <i>license-code</i>	Add the specified license code to the redundant license file. Enclose the license code in quotation marks. You will be prompted for token distribution and for the threshold percentage. When this percentage of the total number of tokens has been consumed on any license server, token borrowing will occur.
-F <i>license-file-name</i>	Add the license codes contained in the specified text file to the redundant license file.
-D <i>feature version</i>	Delete a license code from the redundant license file for the specified feature/version.
-p <i>pool-name</i>	Change the redundant license server pool name to the one specified. May be 8 characters.
-s <i>sequence-#</i>	Change the sequence number of the redundant license file. (A mismatch in sequence numbers triggers transmission of the redundant license file to the other redundant license servers when the license server on which this change is being made is stopped and restarted. Not necessary except in extraordinary circumstances, because this is handled for you automatically.)
-t <i>time-stamp</i>	Change the time-date stamp of the redundant license file. (A mismatch in time-date stamps triggers transmission of the redundant license file to the other redundant license servers when the license server on which this change is being made is stopped and restarted if the sequence number is the same in all of the redundant license files. Not necessary except in extraordinary circumstances, because this is handled for you automatically.)
-R <i>report-filename</i>	Write the redundant license file contents to the specified file.
-C <i>conflict-report</i>	Write any conflicts between token allocation and license servers to the specified file.

Using WlmAdmin to Maintain a Redundant License Server Pool

Once you have started up **WlmAdmin** and selected the license servers you want to monitor you are ready to use its redundant license server features. (For information on starting up **WlmAdmin**, see “WlmAdmin - Administer License Management Activities” on page 96 of “Chapter 5 - Administrator Commands.”)

Note If you are using **WlmAdmin** to shut down a license server or to turn license server token borrowing on or off, you must be the person who

started that license server or, on Windows NT/2000 and UNIX, you must have administrator privileges.

What Does WlmAdmin Do?

For full information on using **WlmAdmin**, see “WlmAdmin - Administer License Management Activities” on page 96 of “Chapter 5 - Administrator Commands.” Here is a brief summary of its functions.

Creating or Editing the Redundant License File

To create a new redundant license file or to edit an existing one, go to the **WlmAdmin Edit** menu and select the **Redundant License File** command. This calls the **Wrlftool** utility. For information on **Wrlftool**, see “Using Wrlftool to Create or Edit a Redundant License File” on page 54.

Note If you are editing an existing redundant license file, to avoid synchronization problems because someone else is adding licenses or changing distribution criteria in the redundant license file while you have it open, we recommend you make a copy of the file under another name, make your changes, and then rename it back to its original name when you are done. For best performance we recommend you edit the redundant license file on the leader license server.

Also note that the changes you make to the redundant license file will not take effect until you copy the changed file to a redundant license server in the pool (or edit it directly on the license server using the precautions discussed above) and then restart that server.

Creating or Editing the Reservation File

The reservation file defines groups of members for which license tokens can be reserved. You can also exclude members (users or computers) from using the tokens. Go to the **WlmAdmin Edit** menu and select the **Reservation File** command. This calls the reservation file editor, **WlsGrMgr**. See “Editing the Reservation File on Windows (WlsGrMgr)” on page 117 of “Chapter 6 - Setting User Options.”

Monitoring License Servers and Licenses

You can also use WlmAdmin to:

- **Monitor license servers** - See what license servers are running, and find out information about them.
- **Monitor licenses** - See information on licenses on the license servers. You can see whether a license is a redundant license, can be used by commuters, how many tokens are in use, and how its tokens are distributed.
- **Monitor users** - See what users are currently using licenses and what users are queued waiting for a license token.
- **Shut down a license server.**
- **Add a license code to a license server** - Add license codes to the memory of redundant and non-redundant license servers.
- **Delete a license code from a license server** - Delete a license code from the memory of a license server.
- **Change token distribution for a license.**
- **Change the license server log file** - Start a new log file for a license server..
- **Enable/disable license token borrowing** - Turn token borrowing on or off for all license servers or for specific licenses.

Using lspool to Maintain a Redundant License Server Pool

lspool is a command-line utility for Windows and UNIX. (On a Windows computer, use **lspool** from within an MS-DOS window.) For those options that modify the redundant license file, you must have write-access to the computer on which one of the redundant license servers resides. You must set the LSFORCE-HOST environment variable on your computer to point to the license server's computer before using **lspool**.

Note When making changes to the redundant license server pool as a whole (for example, when adding or deleting license servers to the pool), for best performance we recommend you select the leader license server to make the changes to.

Tip If you are running on a Windows computer, you may find using the Windows-interface **WlmAdmin** program more convenient than using **lspool**.

The **lspool** options are:

Option	Description
-h	(Help.) Displays list of lspool options.
-a license-server	Add the license server to the redundant license server pool. Specify the host name, IP address, or IPX address of the computer on which the license server resides. Modifies the redundant license file.
-d license-server	Delete a license server from the redundant license server pool. Specify the host name, IP address, or IPX address of the computer on which the license server resides. Modifies the redundant license file.
-l	Display the host name of the leader redundant license server.
-p	Display the list of license servers in the redundant license server pool.
-c feature version dist-crit.	Change the distribution criteria for the specified feature/version tokens. The <i>dist-crit</i> format is: <i>server1:tokens1^server2:tokens2....</i> where <i>server</i> is the host name, IP address, or IPX address of the computer containing the license server. If version is not specified, it must be replaced with empty quotation marks: " ". You must separate the <i>server:tokens</i> arguments with a ^ symbol. Temporary change only in effect until the license server is stopped and restarted; does not modify the redundant license file.
-g feature version	Display the distribution criteria for the specified feature/version. If the version is not specified, it must be replaced with empty quotation marks: " ".
-b feature version OFF ON	Disable or enable token borrowing for the specified feature/version for this license server. If version is not specified, it must be replaced with empty quotation marks: " ". Temporary change only in effect until the license server is stopped and restarted; does not modify the redundant license file.
-B OFF ON	Disable or enable token borrowing for all features/versions for this license server. Temporary change only in effect until the license server is stopped and restarted; does not modify the redundant license file.

Option	Description
-L event OFF ON	Disable or enable logging for the specified event. (Specify the number of the event you want to log from the list below.)

Logging Events

Note If you are using the **lspool -L** option to log events, you must be the person who started the license server or, on Windows NT/2000 and UNIX, you must have administrator privileges.

Events that may be logged using the **-L** option are:

-
- 0 Log all events.
 - 1 Log license server up. On by default.
 - 2 Log election of leader license server.
 - 3 Log heart beat.
 - 4 Log borrowing related event.
 - 5 Log usage information event.
 - 6 Log distribution change.
 - 7 Log license information synchronization event.
 - 8 Log redundant license file distribution.
 - 9 Log license sever down.
 - 10 Log license server addition/deletion.
 - 11 Log license addition/deletion.
-

The events you select will be logged into the license server error log file.

Sample Uses of lspool

To see what license servers are in the redundant license server pool:

Use the **lspool -p** option. You see a list of the license servers as well as their status.

Servers in Pool are :

DISCOVERY	-	UP
EMERALD	-	UP
PLATINUM	-	UP
P3-ORION	-	UP
APOLLO	-	UP

To See How Tokens Are Distributed for a Particular License

Use the **lspool -g** option, specifying the license feature/version you are interested in. For example, to see how the license tokens are currently allocated among the redundant license servers for feature 87, version 1.0, use the command:

```
lspool -g 87 1.0
```

You see a display something like this, showing how many tokens for that feature/version are allocated to each license server:

The Distribution Criterion is:

	Units Allocation	Units Issued	Units Committed
EMERALD :	1	0	0
PLATINUM :	1	0	0
DISCOVERY :	18	18	0
P3-ORION :	17	0	0
APOLLO :	1	0	0
FREE_POOL :	0		

To Distribute Tokens for a Particular License

Use the **-c** option to distribute tokens among the license servers for a particular feature/version. For example, to allocate 5 tokens each to the license servers Emerald, Platinum, and Discovery for feature 87 version 1.0:

```
lspool -c 87 1.0 emerald:5^platinum:5^discovery:5
```


Chapter 4 - Commuter Licensing

One of the most useful advantages of the SentinelLM license management system is that it provides commuter licensing, which allows you to temporarily use a protected application on a portable computer that is not connected to the network.

The most common use of commuter licensing is when a user of your network needs to travel on business and wants to take along the protected application. If the license code for that application has been defined by your vendor to allow commuter licensing, the network user can check out an authorization to use the application for the duration of the trip, and then check the authorization back in on returning.

How Does Commuter Licensing Work?

To check out an authorization, the portable computer (for example, a laptop) on which the application will be used must be attached to the network and have access to a license server containing the license for that application. The application must also be installed on that laptop. After checking out an authorization, the user may disconnect the laptop from the network.

Restricting Commuter Licensing

Commuter licensing uses up the same license tokens as other types of licensing. If you want to restrict the percentage of license tokens on a specific license server that can be used by commuter licensing authorizations, use the **-com** option when you start the license server. (See “Appendix A - Using Environment Variables to Configure a License Server” on page 123 for details on license

server startup options.) This makes sure that commuter licensing doesn't use up all of the license tokens needed by other users of the application on the network.

Commuter Licensing Authorization Checkin Policy

When checking out an authorization, the user may request authorization for a specific number of days (the default is 30). Since the authorization expires after that many days, you may choose to instruct the user not to bother checking the authorization in after returning. On the other hand, if you are in danger of running out of license tokens for that application, you may want to encourage users always to check authorizations back in as soon as possible.

Also, note that the commuter licensing user must check an authorization back into the same license server from which the authorization was checked out.

Locking the Authorization

An authorization is locked to the portable computer using parameters set by your application vendor. These parameters might be hard disk ID, host ID, or a variety of other items. If for some reason you need to change which elements are used to lock the authorization, contact your application vendor for assistance.

Before Your User Checks an Authorization In or Out

Before your network user checks out a commuter authorization, make sure that the user's portable computer:

- Has the protected application installed and ready to use. Remember that the application must be able to run off the network.
- Has network access to the appropriate license server. (You do not need to set the *lshost* file or LSHOST or LSFORCEHOST environment variables to contact a specific license server because your user will be able to request a specific license server when using the commuter licensing utility.)

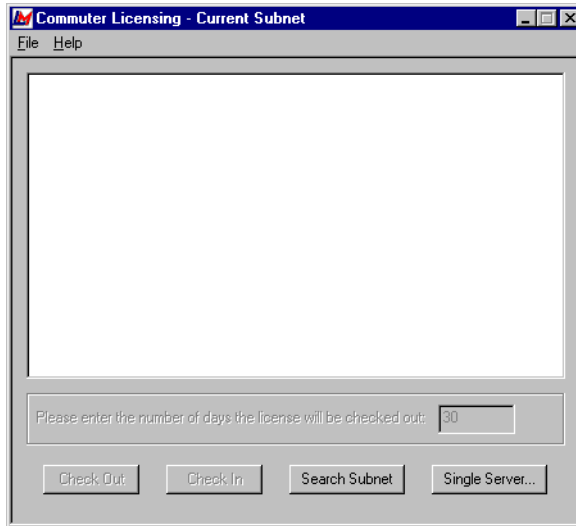
- Is connected to the network.
- Has access to the **Wcommute** or **lcommute** utility. **Wcommute** is a Windows-interface utility (used in an MS-DOS command window); **lcommute** is a command-line utility for UNIX or Windows.

Instructions for Your Network Users

As the network administrator, you will want to give your network users instructions on how to check authorizations in and out. The following sections have been written aimed at your end users. You will want to instruct your users to check an authorization back in as soon as they return, or tell them to just let the authorization expire. (See “Commuter Licensing Authorization Checkin Policy” on page 70.)

Using Wcommute

1. Your network administrator will give you instructions on where to find the **Wcommute** utility. Within Windows Explorer, navigate to the *wcommute.exe* file and double-click it to run it. You see the main **Wcommute** display:

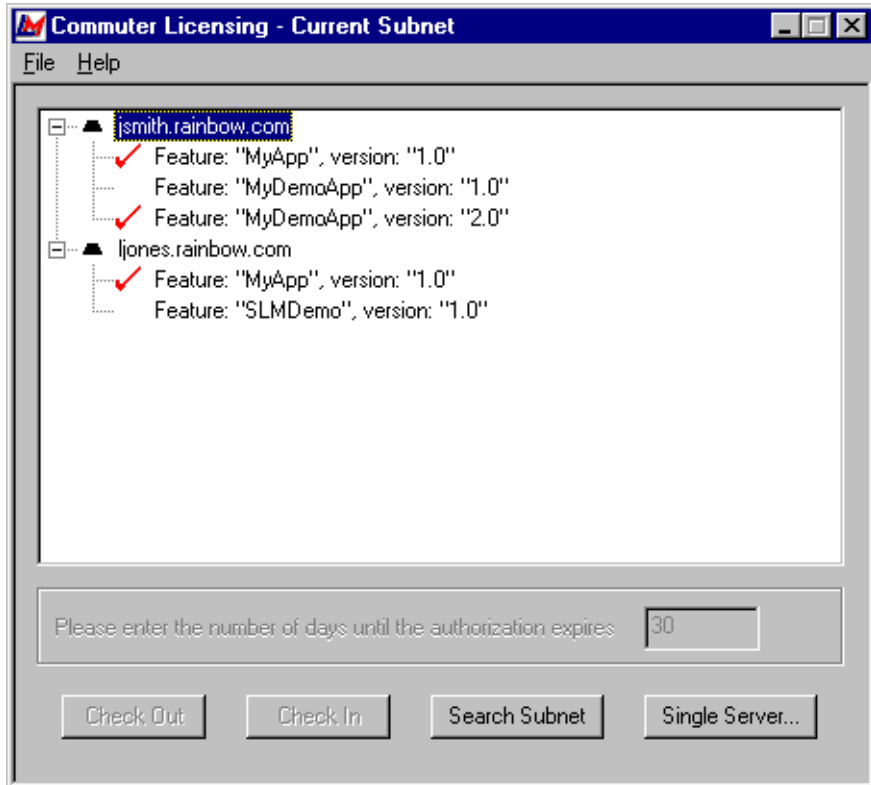


2. To find an authorization for the commuter license you want to use, click **Search Subnet** to see commuter licenses available on all license servers on your part of the network within the given subnet. Or, if there is a specific license server you want to search or want to select a license server outside of your subnet, click **Single Server**.

If you click **Search Subnet**, you may need to wait a minute or two while **Wcommute** searches the subnet for license servers.

If you click **Single Server**, a dialog box displays asking you to enter the name of the license server you want to contact. Enter the license server computer's host name, IP address or IPX address and click **OK**.

3. After selecting the license servers for which you want to see commuter licenses, the **Wcommute** display changes to look something like this:



4. For each license server, you see a list of commuter licenses for which you can check out an authorization. If you see a red check mark next to a commuter license, that means that an authorization has already been checked out to your portable computer. You may not check out an authorization if it is already checked out.

To check out an authorization, click once on a license to select it. The **Please enter the number of days until the authorization expires** text box becomes active. Type a number from one to 30 to select the number of days the authorization will be checked out. By checking out an authorization to use the application, you decrease the number of available uses

of the application for other network users, so specify the minimum number of days that you need.

Next click **Check Out**. Remember the name of the license server from which you obtained the authorization. You will need to check the authorization back into the same license server later.

5. To check in an authorization, look for the license under the license server from which you checked it out. Click once on it in the display to select it. (There should be a red check mark next to it, indicating that you have it checked out.) Click **Check In**.

Note You *must* check the authorization back into the same license server from which you checked it out.

6. To exit **Wcommute**, go to the **File** menu and select **Exit**.

Using **lcommute**

Your network administrator will give you instructions on where to find the **lcommute** utility. From the UNIX or MS-DOS operating system prompt, navigate to the location of **lcommute**, and type the **lcommute** command, specifying the appropriate options listed below.

The **lcommute** options are:

Option	Description
-h	Display list of options.
-c:i	Check in an authorization for a commuter license.
-c:o	Check out an authorization for a commuter license.
-s:license server	Host name, IPX address, or IP address of the license server servicing the commuter license.
-f:feature	Name of feature. (This identifies which license you want to use.)
-v:version	Version. (This also identifies which license you want to use. If the license does not have a version, this can be omitted.)

Option	Description
-d:days	Number of days the authorization will be checked out. Can be 1 to 30. (Only use when you are using the -c:o option.)

Note that if you do not specify any command-line options, **lcommute** will interactively ask you for the information it needs. Here is an example of **lcommute** use:

```
lcommute -s:Sales -c:o -f:99 -d:30
```

In this case, the license server to be contacted is *Sales*, the authorization is being checked out, the feature name identifying the commuter license is *99*, the license does not have a version number (and so it is omitted), and the authorization is being checked out for *30* days.

Chapter 5 - Administrator Commands

Your application may have been delivered with several SentinelLM administrator utilities which are used to install, configure, or to report on the license management system.

Many of these utilities provide information but do not alter the license management system and may be run by an end user without administrative privileges or a detailed knowledge of SentinelLM. A description of these utilities is provided in this chapter. Each description tells when the utility should be used and what level of system knowledge is required to use it.

Table 3-1: Summary of SentinelLM Administrator Utilities

Utility	Description	Where documented
echoid	Obtains information about a computer that will be used to generate a license. This process is called "fingerprinting."	"echoid - Get a Computer's Locking Code" on page 79.
ipxecho	Used to display a computer's IPX network address. Use on the computer on which the license server resides.	"ipxecho - Display the IPX Network Address" on page 80.
lcommute	Command-line utility to check a commuter authorization out or back into the network.	See "Using Iscommute" on page 74.
loads	Used with Windows NT/2000 only. Installs/uninstalls the license server into the System Services registry and starts the service.	"loads - Install the License Server as an NT/2000 Service" on page 81.
lsdecode	Decrypts part of the information in a license code.	"lsdecode - Decrypt and Display a License Code" on page 81.

Table 3-1: Summary of SentinelLM Administrator Utilities (Continued)

Utility	Description	Where documented
Iserv, Iserv9x, Iservnt	Starts the license server.	"Iserv, Iserv9x, Iservnt - Start the License Server" on page 82. Configuration information is also contained in "Appendix A - Using Environment Variables to Configure a License Server" on page 123 and "Chapter 2 - Configuring the License Server" on page 15.
Islic	Installs a license. You need the license code from your application developer.	"Islic - Install/Delete a License Code" on page 84 and "Using Islic to Add Redundant Licenses" on page 51.
Ismail	Used with Windows 95/98 and NT/2000 only. Configures e-mail alerts.	"Configuring E-mail Alerts for Windows" on page 114.
Ismon	Monitors license server transactions. For Windows computers, you may want to use WlmAdmin .	"Ismon - Monitor License Server Transactions" on page 87.
Ispool	Temporarily configures the redundant license server pool.	"Using Ispool to Maintain a Redundant License Server Pool" on page 63.
Isrvdown	Shuts down the license server.	"Isrvdown - Shut Down the License Server" on page 89.
Isusage	Displays information from a license server log file. Can be used to produce CSV files used with Microsoft Access 97 on Windows computers to create license token usage reports.	"Isusage - Display the SentinelLM Usage Log File" on page 89. Also see "Creating License Server Use Reports" on page 93.
Iswhere	Displays the names and/or IP/IPX data of the computers on the network running license servers. For Windows computers, you may want to use WlmAdmin .	"Iswhere - Display a List of License Servers" on page 95.
rlftool	Create the redundant license file.	"Using rlftool to Create or Edit a Redundant License File" on page 59.
Wcommute	Used on Windows95/98 and NT/2000 computers to check an authorization for a commuter license out or back into the network.	"Using Wcommute" on page 71.

Table 3-1: Summary of SentinelLM Administrator Utilities (Continued)

Utility	Description	Where documented
WlmAdmin	Used on Windows95/98 and NT/2000 computers for general SentinelLM administration. Can be used to monitor license server transactions. Gives information on license servers, licensed applications, and application users. Also calls other SentinelLM utilities such as the reservation file editor, the redundant license server pool editor, and the license server shutdown utility.	"WlmAdmin - Administer License Management Activities" on page 96. Also see "Using WlmAdmin to Maintain a Redundant License Server Pool" on page 61.
WisGrMgr	Reservation file editor. Used on Windows 95/98 and Windows NT/2000. Provides an easy way to create a reservation file that reserves license tokens for specific network users or computers, or excludes specific users or computers from using license tokens.	"Setting Group Reservations" on page 115.
Wrlftool	Used on Windows95/98 and NT/2000 computers to create a redundant license file.	"Using Wrlftool to Create or Edit a Redundant License File" on page 54.

echoid - Get a Computer's Locking Code

Format `echoid [-s]`

Platform Windows 95/98, Windows NT/2000, and UNIX.

Who/Where Run from the operating system command prompt by an administrative or application user on a stand-alone, client, or license server computer. The file *echoid.dat* must be in the same directory as the **echoid** program file.

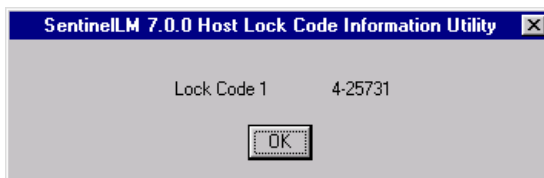
Options The **-s** option displays the locking codes on the command line instead of in a separate message box. This allows you to use redirection to send the output to a text file. For example, on a Windows computer:

```
echoid -s >lock.txt
```

sends the locking code data to a text file named *lock.txt* in the directory from which you used **echoid**.

Description An application or license server may be restricted to operation on certain computers. These restrictions are included in the license codes when they are created. Before a license code with any computer restrictions can be created by your vendor, you must get the locking code for your computer with the **echoid** program. Your vendor will tell you which computers you must run **echoid** on. The **echoid** utility outputs either one or two hexadecimal locking codes depending upon how your vendor has configured it. You will send all locking information to your vendor. Your vendor will use this information, which is specific to your computers, to generate new license codes.

Display The **echoid** display looks like this:



When you use the **-s** option, you see the lock codes separated by semicolons on the command line instead of in a separate message box.

See Also The discussion of the **lslic** utility in “Installing License Codes” on page 7 and “Computer Fingerprints and Locking Codes” on page 2 of “Chapter 1 - Getting Started.”

ipxecho - Display the IPX Network Address

Format ipxecho

Platform Windows 95/98, Windows NT/2000, and UNIX: **ipxecho**.

Who/Where Run by an administrative user on the computer containing the license server.

Options None.

Description Displays the IPX network address. When using the IPX network protocol, the license server host name must be the IPX address of the computer on which the license server resides. Use this utility to display that address. The address is returned in the form of four hexadecimal bytes (network-node address) followed by six hexadecimal bytes (IPX-address), for example:

```
00-00-12-34,00-00-86-1A-23-A3
```

loadls - Install the License Server as an NT/2000 Service

Format `loadls`

Platform Windows NT/2000 only.

Who/Where Run by a user on a license server computer. User must have administrative privileges to load or remove an NT/2000 System Service.

Options None.

Description Installs or uninstalls the SentinelLM license server into the System Services registry.

See Also See “Windows NT/2000 Service” on page 30 of “Chapter 2 - Configuring the License Server.”

lsdecode - Decrypt and Display a License Code

Format `lsdecode [-s license-file] [-e license-configuration-file]`

Platform Windows 95/98, Windows NT/2000, and UNIX:.

Who/Where Run from the operating system command prompt by an administrative or application user on a stand-alone, client, or license server computer.

Options One of the following options may be supplied:

Option	Description
-s <i>license-file</i>	The name of the license file. If this is not specified, lsdecode uses the default file name, <i>lservrc</i> , in the current directory. The location of the license file can also be provided by the environment variable, <code>LSERVRC</code> . You may also specify a text file containing a single license code. You cannot use lsdecode to read a redundant license file (default name <i>lservrff</i>), but can use it to read a redundant license code file containing an uninstalled redundant license code.
-e <i>license-config-file</i>	The name of the configuration file, which may be needed in case readable license strings have been customized by remapping of fixed strings. By default, lsdecode looks for <i>license-file.cnf</i> (for example, <i>lservrc.cnf</i>). The location of the configuration file can also be provided by the environment variable, <code>LSERVRC CNF</code> .

Description The **lsdecode** utility decrypts part of the information in license code strings. This utility can be useful in determining the details of licensing agreements. It also enables you to decipher unknown codes.

lserv, lserv9x, lservnt - Start the License Server

Format (Depends on platform.)

Platform Windows 95/98: **lserv9x**.
Windows NT/2000: **lservnt**.
UNIX: **lserv**.

Who/Where Run from the operating system command prompt by an administrative user on license server computers. For details on user privileges required to start and configure a license server, see “User Privileges” on page 28 of “Chapter 2 - Configuring the License Server.”

Options

Below is a list of the most commonly used options. For a complete list, see “LSERVOPTS - Set License Server Options” on page 126 of “Appendix A - Using Environment Variables to Configure a License Server.” We recommend that you use the LSERVOPTS environment variable to make configuration changes to the license server, since options used at license server startup are in effect only until the license server is restarted.

Option	Description
-z <i>usage-log-file-size</i>	The maximum size of the usage log file. The size can be specified in bytes, kilobytes, or megabytes. For instance, -z 2000 means 2000 bytes, -z 2k means 2 kilobytes, -z 2m means 2 megabytes. The default value is 1 megabyte. Once the maximum size of the file is reached, the license server will create a backup log file unless the -x option has been used; the -x option causes the license server to simply stop logging transactions. For information on log file backups, see “LSERVOPTS - Set License Server Options” on page 126 of “Appendix A - Using Environment Variables to Configure a License Server.”
-b	For Windows 95/98, starts the license server as a background service which will not be terminated when the computer user logs off. The license server runs without displaying a window or message. You must use the Isrvdown utility to stop the license server.
-x	By default, on overflow of the usage log file, the file contents will be moved into a backup file. New usage records are then written to the original file until it overflows again. If the -x option is specified, the file will not be backed up on overflow; instead the license server will simply stop writing further records to the file.
-q	Quiet mode. When this option is specified, the license server will start up quietly without displaying its banner. Unexpected conditions will still be logged as usual. (UNIX only.)

For Windows NT/2000 only, the following system service options are also available:

Option	Description
-H or -h	Display list of Iservnt options.
-X remove	Remove the license server service.

Option	Description
-X install	Install the license server service.
-X start	Start the license server service.
-X stop	Stop the license server service.
-X status	Get license server status.
-X commline	Start the license server service from the command line. If you use this option, it must be the first option on the command line.

Description Starts the license server and sets various license server options. Command line options are designed for temporary changes; these changes are only in effect until the license server is restarted. To make permanent changes, use the LSERVOPTS environment variable.

See Also “Chapter 2 - Configuring the License Server” on page 15. Also see “Appendix A - Using Environment Variables to Configure a License Server” on page 126 for a list of all license server options.

Islic - Install/Delete a License Code

Format `islic [option]`

Platform Windows 95/98, Windows NT/2000, UNIX.

Who/Where Run from the operating system command prompt by an administrative user on a stand-alone, client, or license server computer. Works for stand-alone or network computers. If you are using **Islic** to modify a license file on a remote computer, you must have write-access to that computer.

Options One of the following options must be specified:

Option	Description
-a <i>license-code</i>	Adds a license code to the contacted license server without updating the license file. (Change is temporary until the license server is restarted.)

Option	Description
-A <i>license-code</i>	Adds a license code to the license server's license file (<i>Iservrc</i>) and communicates the change to the contacted license server. (Change is permanent.)
-d <i>feature version</i>	Deletes a feature/version from the contacted license server. (Change is temporary until the license server is restarted.)
-f <i>filename</i>	Adds all the license codes in the supplied file to the contacted license server without updating the license file (<i>Iservrc</i>). (Change is temporary until the license server is restarted.)
-F <i>filename</i>	Adds all the license codes in the supplied file to the license server's license file (<i>Iservrc</i>) and communicates the new information to the contacted license server. (Change is permanent.)
-ad <i>license-code dist-crit</i>	Adds a license code to the redundant license servers without updating the redundant license file (<i>Iservrff</i>). Change is temporary until the license server is restarted. If either the license code or the distribution criteria (how the tokens of that license code will be distributed) is not specified, Islic will prompt for that information.
-Ad <i>license-code dist-crit</i>	Adds a license code to the redundant license servers and updates the redundant license file (<i>Iservrff</i>). Change is permanent. If either the license code or the distribution criteria (how the tokens of that license code will be distributed) is not specified, Islic will prompt for that information.
-fd <i>filename</i>	Adds all the license codes and distribution criteria in the file to the redundant license server's memory. (Change is temporary until all of the redundant license servers are restarted.)
-Fd <i>filename</i>	Adds all license codes and distribution criteria in the file to the redundant license server and updates the redundant license file (<i>Iservrff</i>). (Change is permanent.)
-removeall	Removes all license codes (redundant and non-redundant) from the license server but does not delete them from the license file. (Temporary change.)

Description The **lslic** utility installs or deletes a license. If you do not include necessary information on the command line, **lslic** will ask for the missing information. When you enter a license code, if you are entering an encrypted license code, type only the characters before the # symbol.

For information on specifying the distribution criteria for redundant license codes, see “Specifying Distribution Criteria for Redundant Licenses” on page 53 of “Chapter 3 - Setting up Redundant License Servers and License Balancing.”

lslic can add or delete licenses without stopping and restarting the license server, but the change is only temporary unless you use an option that makes the change to the license file. To use **lslic** to modify a license file or redundant license file for a license server on a remote computer, use the LSFORCEHOST environment variable to point your computer to the computer containing that license server.

See Also “Installing License Codes” on page 7 of “Chapter 1 - Getting Started.”

For information on installing redundant license codes, see “Using lslic to Add Redundant Licenses” on page 51 of “Chapter 3 - Setting up Redundant License Servers and License Balancing.”

For information on using other utilities to add and delete redundant license codes, see “Redundant License Server Pool Utilities” on page 50 of “Chapter 3 - Setting up Redundant License Servers and License Balancing.”

Ismon - Monitor License Server Transactions

Format `lsmon [Poll-interval_in_seconds] [Server-host]`

Platform Windows 95/98, Windows NT/2000, UNIX.

Who/Where Run from the operating system command prompt by an administrative or application user on a stand-alone, client, or license server computer.

Options One of the following options may be supplied:

Option	Description
<i>Poll-interval_in_seconds</i>	Enable periodic update at indicated rate. This option is supported on UNIX only.
<i>Server-host</i>	The name of the computer on which the license server is running.

Description **Ismon** retrieves information about all features currently licensed by the SentinelLM license server and clients using those features. If *Server-host* is omitted, **Ismon** will attempt to talk to the SentinelLM license server on the computer indicated in the LSHOST environment variable or in the *LSHOST* (or *lshost*) file. If the variable or the file does not exist, then it will attempt to contact a license server using the broadcast mechanism. If **Ismon** fails to find a SentinelLM license server, it will display an error message and exit.

On UNIX computers, you can use *Poll-interval-in-seconds* to specify that **Ismon** should keep monitoring and reporting usage activity instead of displaying information once and stopping. If specified, **Ismon** will wait for that many seconds between re-polls of the license server. **Ismon** monitors all licensed applications supported by a license server. If the license server supports protected applications from multiple vendors, then all licenses for all vendors will be displayed. See below for a sample display.

See Also “WlmAdmin - Administer License Management Activities” on page 96.

Ismon Display

The information **Ismon** displays about the licenses on the license server depends on the license. In addition to basic information about the license, you can also see whether the license can be used as a commuter license or if it is a redundant license. You also see information about any users that are using the license; for example:

```
License type           : "Normal License."
Commuter License Allowed : YES
Feature name          : "m_comm"           Feature version      : "1.0"
Max concurrent user(s) : 10
Unreserved keys in use : 2               Available unreserved  : 8
Commuter Keys In Use   : 2
Commuter Keys Left     : 0
Reserved keys in use   : 0               Available reserved    : 0
Soft limit on users    : Unlimited.
License start date     : Fri Jan 01 00:00:00 1999
Expiration date        : License has no expiration.
Log Encryption Level   : 1
Able to issue Commuter Licenses : Yes
App-server locking     : Demonstration license.
Additive/exclusive     : Additive license (number of keys will be added).
Held licenses          : Allowed, hold time set by application.
Key lifetime (heartbeat) : 300 secs (5 min(s))
Users:
  User name            : jsmith
  Host name            : jsmith
  X display name       : local
  Group name           : DefaultGrp
  Status               : Running since Wed Aug 04 17:03:46 1999
  Is Commuter Key      : YES
```

lsrvdown - Shut Down the License Server

Format `lsrvdown host-name`

Platform Windows 95/98, Windows NT/2000, UNIX.

Who/Where Run from the operating system command prompt by an administrative user on a client or license server computer. On Windows 95/98, only the same user who started the license server can stop it. On UNIX, only the user logged in as “root” can use **lsrvdown**. On Windows NT/2000, any user with administrative privileges may use **lsrvdown** unless the license server was started by a user logged in as Administrator, in which case only Administrator can shut the license server down. The computer on which you are running **lsrvdown** must be in the same network domain as the license server computer.

Options *host_name* specifies the name of the computer that is running the license server that you want to shut down.

Description Shuts down the named license server.

lsusage - Display the SentinelLM Usage Log File

Format `lsusage [options]`

Platform Windows 95/98, Windows NT/2000, UNIX.

Who/Where Run from the operating system command prompt by an administrative or application user on a stand-alone, client, or license server computer.

Options One or more of the following options must be specified::

Option	Description
-l <i>log-file</i>	Selects the license server log file to display. Must be specified. If this is the only option, the summary will be displayed on the screen.
-c <i>CSV-format-file</i>	Specifies the name of an output file to which the output will be written in Comma-Separated Values (CSV) format. This CSV file can be used with Microsoft Access to produce graphical reports; for details, see “Creating License Server Use Reports” on page 93.
-f <i>feature1[,version1]: feature2[,version2]...</i>	Specifies feature(s) and version(s) to be included in the output. Note that <i>version</i> is optional. If this option is not supplied, the output will contain all features and versions in the log file.
-m <i>start-month</i>	Specifies the first month (1-12) of the time period to be included in the output. If not specified, then the -a option is ignored and January 1 is assumed for the start month and start day.
-a <i>start-day</i>	Specifies the first day (1-31) of the month to be included in the output. For example, if you have specified March as the start month, specifying a start day of 1 means that the output will be for a time period beginning with March 1. If not included, and the -m option is included, a start day of 1 is assumed.
-y <i>start-year</i>	Specifies the first year of the time period to be included in the output. Use four digits to specify the year. If not included, then -m and -a are ignored and the output begins with the first data in the file.
-M <i>end-month</i>	Specifies the last month (1-12) of the time period to be included in the output. If not specified, then the -A option is ignored and the default value of December 31 is assumed.
-A <i>end-day</i>	Specifies the last day (1-31) of the month to be included in the output. If not included, and the -M option is included, an end day of 31 is assumed.
-Y <i>end-year</i>	Specifies the last year of the time period to be included in the output. Use four digits to specify the year. If not included, then -M and -A are ignored, and the output ends with the last data in the log file.
-h	Displays a list of all options that can be used.

Description `Isusage` displays a summary of application usage, providing information on license transactions contained in the license server usage file. See below for a sample.

See Also “Setting Usage Logging” on page 23 of “Chapter 2 - Configuring the License Server.” Also see “License Server - Commonly Used Variables” on page 126 of “Appendix A - Using Environment Variables to Configure a License Server.”

Isusage Display

The `Isusage` display looks something like this:

```

                                Application Usage Summary
                                -----
Feature name/ %age    %age    Ttl    %age    %age    Min. App.  Avg. App.  Max. App.
version      Denied  Issued  Keys   Queued  QReq,     Duration  Duration  Duration
              (Minutes) (Minutes) (Minutes)
88 v1.0      0.00   100.00  1      0.00   0.00      0.92      0.92      0.92
**** LOG REPORT FOR Sessions: 5

```

where:

Element	Description
Feature name/Version	Identifies the license for which this entry was made.
%age Denied	The percentage of requests for this license that were denied (usually because the hard limit of the license had already been reached), or if license queuing is enabled, this is the percentage of queued requests denied.
%age Issued	The percentage of requests for this license that were granted.
Ttl Keys Issued	The number of tokens for this license that were issued.
%age Queued granted	The percentage of queued license requests that were granted.
%age Qreq.	The percentage of license requests that were placed in the license queue. (License requests are queued only if license queuing is enabled for this license.)
Min. App. Duration	The minimum number of minutes the application for this license was in use.
Avg. App. Duration	The average number of minutes the application for this license was in use.
Max. App. Duration	The maximum number of minutes the application for this license was in use.
LOG REPORT FOR Sessions: x	The session numbers for this license server that were logged in this file.

Warning **lsusage** will not display license transactions that are encrypted to level 3 and 4. For information on encrypted log file transactions, see “Encrypting License Server Log File Entries” on page 26 of “Chapter 2 - Configuring the License Server.”

Note If the SentinelLM log file contains characters that are not 7-bit ASCII (for example, if a user name contains multi-byte characters such as Japanese Kanji characters), those characters will not be viewable when looking directly at the log file. However, if you use the **lsusage -c** option to create CSV-format output from the log file, you will be able to view the multi-byte characters if you view the CSV-format file with an appropriate text editor on an operating system that displays the multi-byte language. You can also create Microsoft Access reports from the CSV-format file that can be viewed on a computer using the appropriate multi-byte operating sys-

tem. (For information on creating reports from log file CSV-format output, see “Creating License Server Use Reports” on page 93.)

Creating License Server Use Reports

SentinelLM provides the ability on Windows computers to use Microsoft Access 97 to create graphical reports of license server use. To use this feature, you will need to use the **Isusage -c** option to create a comma-separated values (CSV) file from the license server log file. See the section above for information on this option. By use of other **Isusage** options, you can create an output file with use data for specific features and for specific time periods. Once you have created the CSV file, here is how to create graphical reports:

Note You must have Microsoft Access 97 on the computer on which you want to run reports. Also, keep the number of features for which you are generating reports to a reasonable number (15 to 20) or the reports may not be very easy to read because there simply won't be room for a clear display.

To Import the CSV File

1. Within Microsoft Access, open the MDB file provided with SentinelLM: *ReportDB.mdb*.
2. Select the Table tab.
3. Click **New**.
4. Select **Import Table**.
5. Select the CSV file you created with **Isusage** and click **Import**.

Tip In Microsoft Access, not all file types are installed by default. If you do not have the text file type installed, run the Access Setup program, click **Add/Remove**, then select **Data Access** and **Change Option** to install the text file type.

6. The **Import Text Wizard** is displayed. Continue to click **Next** until it asks “Where would you like to store your data?”

7. Select **In an existing table** and select the table *Lservlog*. (If instead of adding to an existing *Lservlog* table you want to create a new one, delete any rows that are already in the *Lservlog* table before importing new data.)
8. Click **Finish**. Don't forget to keep the table name set to *Lservlog*.

Creating Reports

Once the log file use data is imported into the *Lservlog* table as discussed above, you can now create reports for that data.

1. Within Microsoft Access, open the *ReportDB.mdb* file. You will immediately see the SentinelLM report selection screen:



2. You can now select the date range of the report by specifying **Start Date** and **End Date**.
3. Now select the type of report you want to create:
 - Total number of tokens granted per feature
 - Maximum number of tokens granted per feature
 - Average number of tokens created per feature
 - Total usage time per feature
 - Total number of tokens denied per feature
 - Percentage of licenses denied per feature

Tip When using **lsusage** to create the CSV file that is the input to these reports, you can limit the data to usage information for specific features and versions.

lswhere - Display a List of License Servers

Format `lswhere [option]`

Platform Windows 95/98, Windows NT/2000, UNIX.

Who/Where Run from the operating system command prompt by an administrative or application user on a stand-alone, client, or license server computer.

Options You may specify one of the following options (**-d** is the default):

Option	Description
-d	Displays details on the license servers found on the network. See below for an example.
-r	Displays just the IP or IPX addresses of the license servers found on the network (resolves the host names).

Description Use **lswhere** to display the network names of the computers running the license server. By default, **lswhere** displays for each license server the address of the computer on which that license server is running as well as its host name.

Example When using **lswhere** with the **-d** option you see the following information for each license server found on the network:

```
Server Address: 10.10.32.100
Server Name: Sales
Is Redundant: YES
Total Clients: 0
Protocols: Both TCP/IP and IPX
```

Where *Server Address* may be an IPX or IP address; *Server Name* is the host name of the computer on which the license server runs; *Is Redundant* tells you if the license server is redundant or non-redundant; *Total Clients* tells how many clients are being serviced by this license server at this time; and, *Protocols* indicates which communication protocols can be used to communicate with the license server.

WlmAdmin - Administer License Management Activities

Format WlmAdmin

Platform Windows 95/98, Windows NT/2000.

Who/Where Run by an administrative user on a stand-alone, client, or license server computer. Using Windows Explorer, navigate to the location of the *WlmAdmin.exe* file, and double-click the file to start the program. (The exact location of this file depends on your application vendor.)

Options None

Description Provides information on licensing activities. Includes information on license servers detected, details on active licenses, and information on licensed users.

WlmAdmin also performs a variety of administrative functions. See below for more information.

See Also “Using WlmAdmin to Maintain a Redundant License Server Pool” on page 61 of “Chapter 3 - Setting up Redundant License Servers and License Balancing.”

What Does WlmAdmin Do?

WlmAdmin performs a variety of licensing administrative functions.

Monitoring Licenses and License Servers

If you are not using redundant license servers, you can use **WlmAdmin** to:

- See what licenses are being serviced by which license servers and see information on those licenses. You will see quite a bit of information on each license: whether it can be a commuter license, what feature/version is licensed, how many users are using the license, the license expiration date, whether the license is a redundant license, how many tokens have been reserved, and so on. See the display below for a sample.
- See information on license servers.
- For each license, see what users have acquired that license and what users are in a queue waiting for the license.

If you are using redundant license servers, you will be able to use **WlmAdmin** as an informational tool as discussed above, but will also be able to:

- Add or delete a redundant license server from the redundant license server file.
- Display a list of all license servers in the redundant license server pool and display the name of the license server leader.
- Disable and enable token borrowing for all licenses or for particular licenses.
- Change or display distribution criteria for specific licenses.

Editing the Reservation File

A reservation file defines groups of users or computers for whom license tokens can be reserved. The reservation file also can be used to exclude specific users or computers from using certain licenses. From the **Edit** menu of **WlmAdmin**, you can select the **Reservation File** command to call the **WlsGrMgr** utility to create a new reservation file or edit an existing one. For information on **WlsGrMgr** and reservation files, see “Editing the Reservation File on Windows (Wls-GrMgr)” on page 117 of “Chapter 6 - Setting User Options.”

Editing the Redundant License File

The redundant license file defines which license servers make up the redundant license server pool as well as how many tokens of which licenses are distributed among those license servers. From the **Edit** menu of **WlmAdmin**, you can select the **Redundant License File** command to call the **Wriftool** utility to create a new redundant license file or edit an existing one. For information on **Wriftool** and redundant license server pools, see “Using Wriftool to Create or Edit a Redundant License File” on page 54 of “Chapter 3 - Setting up Redundant License Servers and License Balancing.”

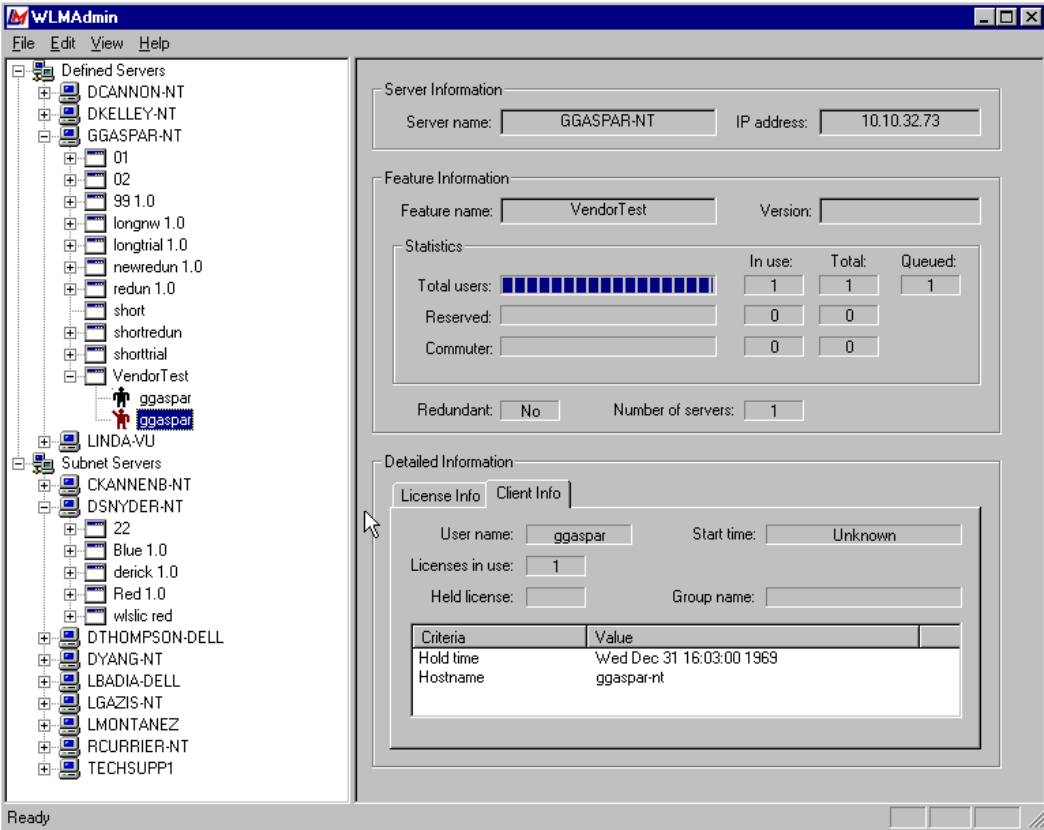
Shutting Down the License Server

In the license server pane of the main **WlmAdmin** window, right-click on the license server you want to shut down and select **Shutdown Server** from the menu. For information on privileges needed to shut down a server, see “User Privileges” on page 28 of “Chapter 2 - Configuring the License Server.”

Using WlmAdmin to Monitor Licenses and License Servers

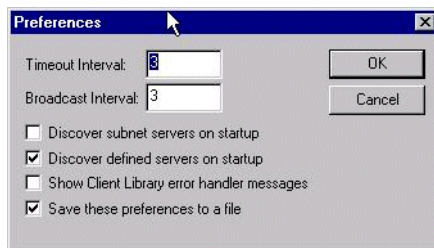
The first time you start **WlmAdmin**, it shows an empty screen, but as soon as you have selected license servers to monitor, **WlmAdmin** displays information

about them. You can also see information about licenses on the license servers and the users who use them.



Setting WlmAdmin Preferences

You can choose what **WlmAdmin** displays when it starts. From the **Edit** menu, select the **Preferences** command:



- To always show the license servers you have defined (see below), select the **Discover defined servers on startup** check box.
- To automatically discover and display the license servers on your subnet, select **Discover subnet servers on startup** check box.

Be sure to select **Save these preferences to a file** to save your choices.

Other preference settings you can choose are:

- **Show client library error handler messages** to pass license management error messages from SentinelLM through to **WlmAdmin**.
- **Broadcast Interval** to select the number of seconds between broadcasts to the network to discover license servers.
- **Timeout Interval** to select the number of seconds at which **WlmAdmin** will stop trying to contact a license server. If you have a large network or a slow one, you might want to increase this number if you find **WlmAdmin** has difficulty retrieving information from license servers.

Selecting the License Servers

The first step to using **WlmAdmin** is to select what license servers you are interested in monitoring.

To Create a List of Defined License Servers

1. Go to the **Edit** menu and select the **Defined Server List** command.
2. Type a license server name and click the **Add** button to add a server to the list. If you are not sure of the license server computer's host name, you can wait for **WlmAdmin** to display license servers in the Subnet Servers area of the left-hand pane of the main window to see what license servers are available on the network.
3. Once a license server is defined, you can add more or select a server and click **Remove** to remove it from the list.
4. Click **OK** to return to the main **WlmAdmin** screen.

To Discover All License Servers on Your Network Subnet

- Go to the left-hand pane of the main **WlmAdmin** window and double-click **Subnet Servers**. **WlmAdmin** will broadcast to the network to discover all license servers and display them here.

To Refresh the Network Display

- Right-click on Defined Servers or Subnet Servers and select **Refresh** to tell **WlmAdmin** to display new information on that group of license servers. You may also select **Refresh All** from the **View** menu to refresh the display for all license servers shown.

License Server-Specific Actions

The following actions affect license servers:

To See Information on a License Server

- Click the name of a license server in the left-hand pane of the main window. In the right-hand pane of the display, you see the IP or IPX address of the computer containing the license server and its host name.

To Add or Remove a License Server from the Defined Server List

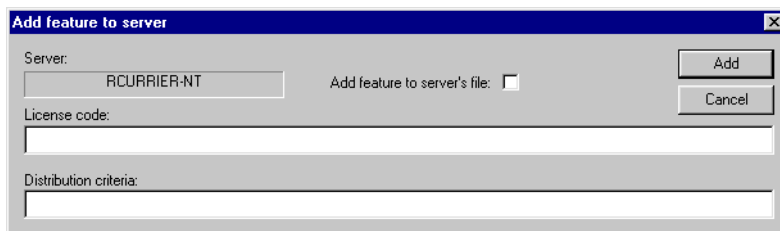
- Right-click the name of a license server in the left-hand pane of the main window. From the menu select **Add to Defined List** or **Remove from Defined List**.

To Shut Down a License Server

- Right-click the name of a license server in the left-hand pane of the main window. From the menu select **Shutdown Server**.

To Add a License Code to the License Server from the Keyboard

1. Right-click the name of a license server in the left-hand pane of the main window. From the menu select **Add Feature**. To add the license by typing in the license code, select **From a String**.



2. Type the license code in the **License code** box.
3. If this is a redundant license code, enter the token distribution criteria in the **Distribution criteria** box in this format:

Server1:tokens1^Server2:tokens2...

where *Server* is an IP address, IPX address, or host name of the computer containing a redundant license server and *tokens* is the number of license tokens to be allocated to that redundant license server. Note that all server:tokens arguments must be separate with a ^ symbol.

4. To permanently add the license code to the license server license file as well as temporarily adding it to the license server memory (only in effect

until the license server is rebooted), select the **Add feature to server's file** check box.

5. Click **Add**.

To Add a License Code to the License Server from a File

1. Right-click the name of a license server in the left-hand pane of the main window. From the menu select **Add Feature**. Select **From a File and To Server** (to add the license code temporarily to the license server memory, only in effect until the license server is rebooted) or **To Server and its File** (to add the feature both temporarily to the license server memory and permanently to its license file).
2. Now select the license code file from the standard file selection window.

To Change the Log File for a Selected License Server

- To begin a new log file for a selected license server, right-click the license server and select **Change log file name**.

To Configure a Redundant License Server File

- From the **Edit** menu, select **Redundant License File** to start the **Wrflftool** utility. For information on using **Wrflftool**, see “Using Wrflftool to Create or Edit a Redundant License File” on page 54 of “Chapter 3 - Setting up Redundant License Servers and License Balancing.”

Note In order to change the redundant license file, you must have write-access to the computer on which the file resides.

To Disable/Enable Token Borrowing for a License Server

- Right-click a license server and select **Enable All Token Borrowing** or **Disable All Token Borrowing**. (You can only use this function for a redundant license server.) Disabling token borrowing for this license server means that this license server will not get more tokens from another redundant license server when it runs out. This affects all licenses on this license server. To turn token borrowing on or off for a

specific license, see “To Disable/Enable Token Borrowing for a License” on page 107.

License-Specific Actions

The following actions affect individual licenses.

To See Information on a License

1. In the left-hand pane of the main **WlmAdmin** window, click the + sign to the left of a license server to see what licenses it is servicing. (Once you have expanded the license server information, you can click the - sign to collapse the information again.) If a license server does not have a + sign next to it, it does not currently have licenses installed.
2. Click the name of a license. In the right-hand pane of the display you see information on the characteristics of the license. The exact information you see depends on the license.

In addition to basic information, such as how many concurrent users can use the license, when it expires, what users currently are using the license, the display can include whether the license can be used for computer licensing, and if the license is a redundant license. If the license is a redundant license, you will see how its tokens are distributed and what license servers can service it. Here is a sample display.

Server Information

Server name: IP address:

Feature Information

Feature name: Version:

Statistics

	In use:	Total:	Queued:
Total users: <input type="text"/>	<input type="text" value="0"/>	<input type="text" value="10"/>	<input type="text" value="0"/>
Reserved: <input type="text"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	
Commuter: <input type="text"/>	<input type="text" value="0"/>	<input type="text" value="10"/>	

Redundant: Number of servers:

Detailed Information

License Info | Client Info | Redundancy Info

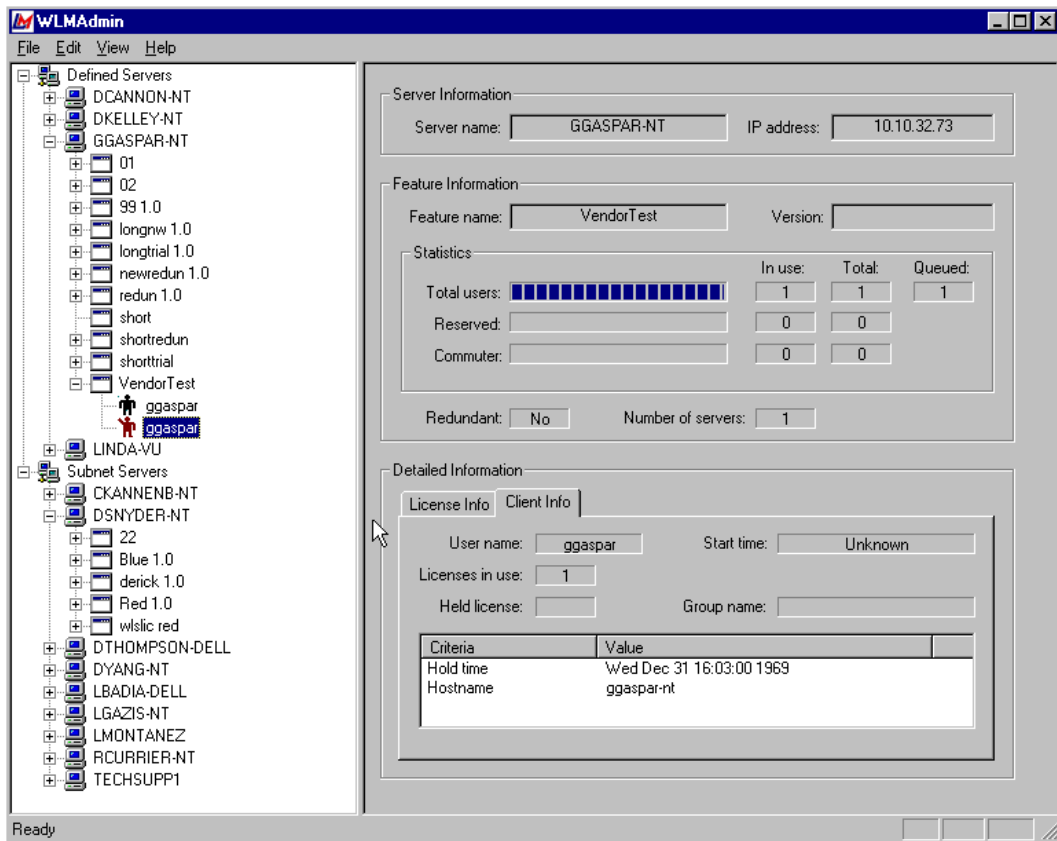
Leader server: Majority rule:

Server	IP Address	Distribution	In use	Status
DCANNON-NT	10.10.20.27	10	0	Unknown
DSNYDER-NT	10.10.20.28	10	0	Active
RYUSOFF	10.10.20.25	30	0	Unknown

3. Double-click a license to see if any users are using the license. (Once you have expanded the license information, you can click the - sign to collapse the information again.)

Tip If a user symbol is red, that user is in a queue waiting for an available license.

- Click the name of a user. In the right-hand pane of the display you see: user name and host name of the computer the user is on. If this user is in a reservation group, you see the name of the group. You also see information on how long the user has been using this license.



To Delete a License from the License Server

- To delete a license from a license server, right-click the license name and select **Remove Feature**. The license will be removed from the license server's memory until the license server is rebooted.

To Change the Token Distribution for a License

- Right-click a license name and select **Change Distribution**. (You can only use this function for a redundant license code.) You see a display something like this:

Distribution Criteria

Feature information
 Feature: Version:

Distribution
 Leader:

Server name:	Licenses:	Total licenses:
1: <input type="text" value="DSNYDER-NT1"/>	<input type="text" value="10"/>	<input type="text" value="50"/>
2: <input type="text" value="RYUSOFF"/>	<input type="text" value="30"/>	
3: <input type="text" value="DCANNON-NT"/>	<input type="text" value="10"/>	
4: <input type="text" value=""/>	<input type="text" value="0"/>	Allocated licenses: <input type="text" value="50"/>
5: <input type="text" value=""/>	<input type="text" value="0"/>	
6: <input type="text" value=""/>	<input type="text" value="0"/>	
7: <input type="text" value=""/>	<input type="text" value="0"/>	
8: <input type="text" value=""/>	<input type="text" value="0"/>	
9: <input type="text" value=""/>	<input type="text" value="0"/>	
10: <input type="text" value=""/>	<input type="text" value="0"/>	
11: <input type="text" value=""/>	<input type="text" value="0"/>	

For information on distributing tokens among redundant license servers, see “Chapter 3 - Setting up Redundant License Servers and License Balancing” on page 53.

To Disable/Enable Token Borrowing for a License

- Right-click a license and select **Enable Token Borrowing** or **Disable Token Borrowing**. (You can only use this function for a redundant license code.) Disabling token borrowing for this license means that when the license server runs out of tokens for this license, it will not get more tokens from another redundant license server. To turn token bor-

rowing on/off for all licenses on the license server, see “To Disable/Enable Token Borrowing for a License Server” on page 103.

Exiting WlmAdmin

To leave **WlmAdmin**, go to the **File** menu and select **Exit**.

Chapter 6 - Setting User Options

The following advanced options are available for SentinelLM users:

- Alerts
- Group allocation

An alert lets you know about license events, such as when a license reaches its expiration date. You specify alerts using a configuration file. There are seven alerts including hard and soft limits that can be generated. The reporting mechanisms are e-mail and script. For more information about alerts, see “Setting User Alerts” on page 109 below. Alerts are most frequently used with a network license server. This capability is also available, however for a stand-alone license application.

You can also use group reservations to restrict the right to run applications to particular users, computers, or groups. For more information, see “Setting Group Reservations” on page 115. The group reservations capability is only meaningful with a network license server.

Setting User Alerts

You can specify/configure alerts through a simple file interface.

The alerts handled are:

- Soft limit exceeded
- Hard limit reached (the maximum number of licenses allowed)
- License issued

- License returned
- License denied (occurs on the next request after the hard limit has been reached)
- License time-out
- License expiration date reached

The alert reporting mechanisms are:

- E-mail
- Script

Alert Specification

To enable alerts, you need a configuration file with information on the license codes in the license code file.

Configuration information can be readable license remap statements, alert action statements, etc. The license server is the only entity that deals with alerts. The configuration file is read by the license server and **lsdecode**, which ignore statements in the configuration file that they are not interested in. If the configuration file does not exist, no alerts will be reported. The file can contain the information on what to do with the alerts. If the configuration file is changed while the license server is running, the license server will need to be restarted in order for the changes to take effect.

The configuration file, *lservrc.cnf*, is a general-purpose configuration file associated with a particular license file. The environment variable `LSERVRC CNF` can specify the path to the configuration file. The path for `<LicenseFile>.cnf` is constructed from the license file path the user is using. *LicenseFile* can be specified using existing methods such as the license server startup `-s` option, or the `LSERVRC` environment variable. It is not an error for the configuration file to be missing.

The configuration file is broken into sections, headed by feature and version:

```
[feature_name1 feature_version1]
remap-statement-11
remap-statement-12
. . .
alert-action-11
alert-action-12
. . .
[feature_name2 feature_version2]
remap-statement-21
remap-statement-22
. . .
alert-action-21
alert-action-22
. . .
```

For more detailed information about the configuration file format, contact your vendor.

Each alert-action looks like:

```
alert-type=reporting-mechanism1 ON/OFF
reporting-mechanism2 ON/OFF
```

Each section can contain further customizing statements:

```
SCRIPT=script-path
EMAIL=email-addr1 email-addr2...
```

alert-type can be:

- softlimit—Soft limit exceeded
- hardlimit—Hard limit reached
- appstart—License issued
- appstop—License returned
- denied—License denied
- apptimeout—License time-out

- expired—License expiration date reached

Note appstart and appstop may generate a lot of network traffic.

An expired alert will occur when either condition occurs:

- An expired license is added to the license server and the server does not already have an unexpired license for the same feature.
- license expires due to passage of time.

reporting-mechanism can be:

- script—The script given after SCRIPT= will be invoked
- email—E-mail will be sent to recipients given after EMAIL=

Line continuation is not supported, so all e-mail addresses must fit on one long line. The maximum length of a line is 512 characters.

An example configuration file is shown below:

```
# This file remaps the keyword "LONG" to "lng" for all readable
# licenses for feature DOTS (any version).
# It also provides alert actions for all features licensed
# by the license server.
# Note that the alert actions that apply to all non-DOTS
# features do NOT apply to DOTS since DOTS has its own section.
[DOTS *]      # For DOTS, we generate only the denied alert
LONG=lng      # Remap statement
denied = Script OFF  Email ON
EMAIL = jsmith@Engr1

[* *]        # For non-DOTS apps, we generate most of the alerts
softlimit = SCRIPT OFF EMAIL ON  # Alert action statement
hardlimit = SCRIPT On  EMAIL On
denied = Script on  Email ON      # This better not happen
apptimeout = script on  Email off
expired = SCRIPT on  email on
EMAIL = mark@jupiter
SCRIPT = /usr/local/etc/lsalertscr
```

One section of the file can specify only a single alert script and a single e-mail address. For all alerts enabled in one section, the same alert script will be invoked.

SCRIPT

SCRIPT will receive the following command-line arguments:

```
SCRIPT alert-type feature,version user-name client-host-name limit
```

If a certain argument is not applicable to the alert that occurs, the value NONE is passed in place of that argument.

To be more specific, the values for limit will be:

- `softlimit`—Soft limit
- `hardlimit`—Hard limit
- `appstart`—Number of licenses checked out after the event
- `appstop`—Number of licenses checked out after the event
- `denied`—Number of licenses checked out when the event occurred
- `apptimeout`—Number of licenses checked out after the event
- `expired`—NONE

If script mechanism is ON, and there is no SCRIPT statement in the same section, script mechanism will be turned off.

If e-mail mechanism is ON, and there is no EMAIL statement in the same section, e-mail mechanism will be turned off.

EMAIL

E-mail alerts are available on both Windows and UNIX computers, but are configured differently for each.

Configuring E-mail Alerts for Windows

On Windows computers, the license server calls the **lsmail** utility to send e-mail alerts using SMTP. Here is how to use **lsmail**:

1. Set up the alert information section of the license server configuration file and use the EMAIL statement to enable e-mail alerts.

The file *lsmail.exe* should be in the directory in which the license server resides.

2. Run **lsmail**. The first time it is run, it will ask for configuration information: the SMTP server address (the mail server), the port number to be used (default is 25), and the sender name (default is SentinelLM).
3. **lsmail** will automatically be called by the license server to send e-mail when the alert action specified in the license server configuration file occurs.

If you want to run **lsmail** manually, its options are:

-t <i>recipient-name</i>	E-mail address of the person who will receive the alert message.
-m <i>message-text</i>	Text of the alert message.
-s <i>subject</i>	Subject (if your e-mail program accepts a subject argument).

To test **lsmail**, at the DOS prompt type (specifying your e-mail address):

```
lsmail -t "user@company.com" -m "This is a test" -s "Test"
```

Configuring E-mail Alerts for UNIX

The mail sending program to use can be customized via the MAILPGM environment variable. If MAILPGM is not set, the first program found from the following locations will be used. Search the path in the license server's start-up environment (PATH environment variable). Any command/shell aliases will be ignored.

- “/usr/bin/mailx”

- “/usr/ucb/mail”
- “/usr/bin/mail”
- “/bin/mail”

If no program is found via search or MAILPGM, the e-mail mechanism will be off.

The mail program will be invoked in a manner equivalent to:

```
MAILPGM email-recipient-list text-of-alert-message
```

No subject is provided.

If your mail program accepts a subject via the `-s` option, you could use:

```
MAILPGM= /usr/ucb/mail -s "Alert message"
```

to receive a subject for outgoing alert email.

The configuration file is read once per feature and version on the first addition of that feature (not necessarily at the license server start time), and the statements for that feature and version are cached. If you change the entries applicable to a feature, you should restart the license server.

Setting Group Reservations

Group reservations give you the ability to exercise local administrative controls by associating a series of groups with each feature and reserving for each group a certain number of license tokens. Any license tokens not specifically reserved can be used by anyone.

Including/Excluding Users or Computers

Selected users can be allowed to run certain applications or can be prevented from accessing applications. This can be used as an additional security measure.

Certain computers can also be allowed to run or prevented from running applications. This can help to balance application usage when you want to prevent users from running applications on heavily loaded computers or file servers.

How the Reservation File Works

The reservation file tells SentinelLM how many license tokens are reserved for members of different groups. For example, the Sales department could be a group with ten members for which 15 license tokens are reserved. The members might be network users or specific computers on the network. The reservation file might also tell SentinelLM that certain members of the group are excluded from taking license tokens.

The groups defined must be mutually exclusive. Different groups for the same feature should not have common users or computers. The number of license tokens reserved for a feature cannot exceed the number of concurrent users specified in the license code for that feature.

When the license server receives a request, it checks whether the user making the request belongs to a group. If so, and license tokens are available for that group, the license server will issue the token(s) and decrement the token count on the license server. Otherwise, requests will be serviced with tokens that have not been reserved and that are available. Users for whom tokens have been reserved have first call on available tokens.

At start up, the license server consults the environment variable, `LSRESERV`, for the path and file name of the reservation file. If the variable is not set, the local directory is searched for the file, `lsreserv`. If no file is found, the license server assumes that no reservations exist.

Reservation File Characteristics

The following apply to groups:

- A maximum of 256 groups with 256 members (user names or computer IDs) each are supported.
- Groups must be mutually exclusive.

- Different groups for the same feature should not have common users or computers.
- Group names and member names cannot exceed 64 characters each.
- The number of license tokens reserved for a feature cannot exceed the number of concurrent copies specified in the license code for that feature.
- If you want to exclude all members of a group from using license tokens for a feature, you can specify 0 tokens.
- If the number of license tokens reserved is greater than the number of concurrent copies specified for the feature in the license code, the number of tokens specified in the license code will be used.

Editing the Reservation File on Windows (WlsGrMgr)

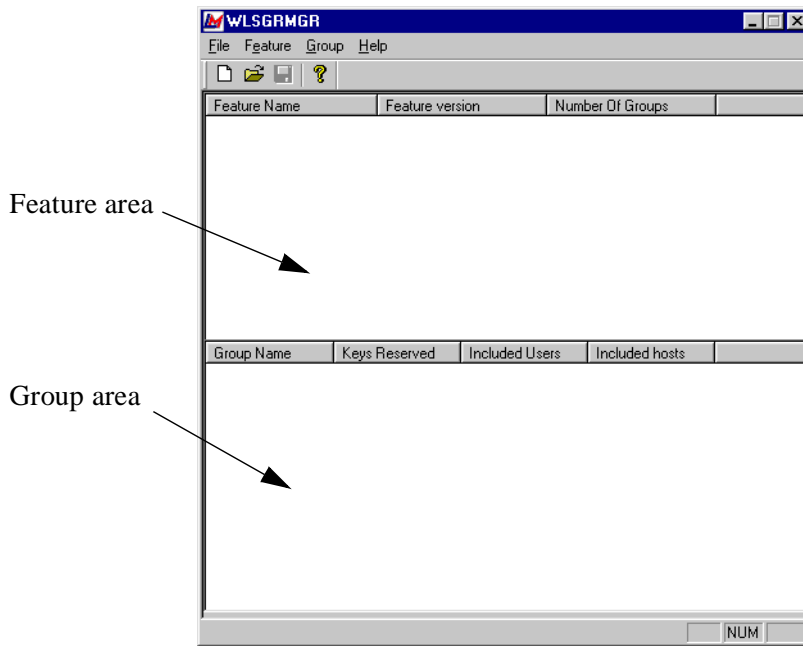
On Windows computers, the **WlsGrMgr** utility can be used to create or edit a reservation file.

Tip You can call **WlsGrMgr** from within **WlmAdmin** by using the **WlmAdmin Edit | Reservation File** command.

Note that if you create a new reservation file in a directory containing an existing reservation file, **WlsGrMgr** will append a number to the name of the new reservation file to avoid overwriting the existing one (for example, *lreserv1*, *lreserv2*, and so on).

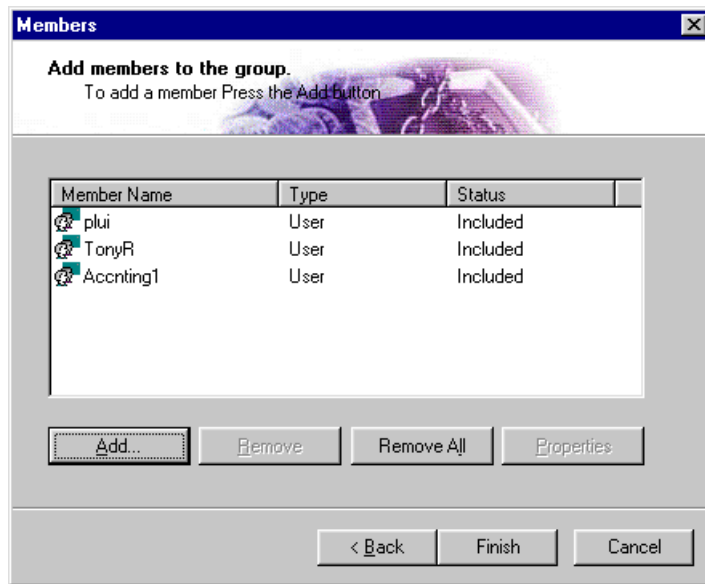
1. Using Windows Explorer, navigate to the location of the *wlsgrmgr.exe* file. Double-click the file to start the program. (The exact location of the file depends on your application vendor's installation directory.)

You now see the main **WlsGrMgr** screen:



2. To make a new reservation file, select the **New** command from the **File** menu. Or use the **Open** command on the **File** menu to edit an existing reservation file.
3. To add definitions for who can use license tokens for specific licenses, right-click in the **Feature Name** area of the screen and select **Add** from the menu. The **New Feature Wizard** appears.
4. Click **Next**. The wizard asks for the feature name and the version (optional) of the license. After entering that information, click **Next**.
5. Now the wizard asks for a group name and the number of tokens to reserve for that group.

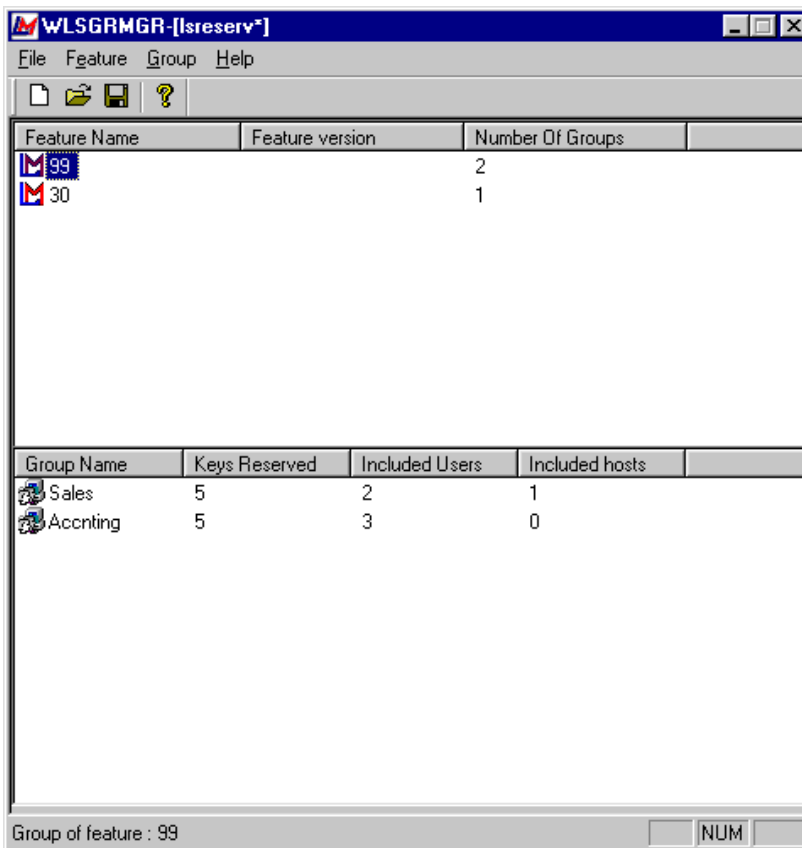
6. Now you see a box where you can add the members of the group.



7. Click **Add** to add a new member. You are asked to enter the name of the member, and whether it is a user (if so, use the user's login name) or a machine (use the name of the computer on your network). You are also asked if you want to include or exclude this user or machine. Click **OK** to finish the definition.

Once one or more users are defined, you can click **Remove All** to remove all of them and start over or select a specific member and click **Remove** to remove just that member.

8. Click **Finish** to go back to the main screen.
9. At the main screen, you can see the results of adding groups and members:



- After at least one feature has been defined, you can define another group to use that feature. Right-click in the **Group** area and select **Add** from the menu to add a new group. Do not add the same users or computers to different groups using the same feature.

Manually Editing the Reservation File

For UNIX computers, you will need to edit the reservation file manually rather than using the **WlsGrMgr** utility discussed above.

A group specification consists of:

- The name of the feature for which the reservation applies;
- The name of the group;
- The number of license tokens reserved for that group;
- And the login names of users or host IDs of computers that belong to that group that will included or excluded from using the feature.

Group reservations should be entered according to the following format, with one group per line:

```
feature_name[ ,ver ]:group_name:num_of_licenses:{ user_name | computer }
```

One or more of *user_name* and/or *computer* may be specified, but at least one value must be specified in the last field. Version number is optional. If no version number is specified, only the feature name is used.

The characters \$ and ! have special meaning: \$ indicates the computer name, and ! indicates a logical NOT.

For example, two users, one computer in *group 1*:

```
stars:group1:2:tom bryant $athena
```

One user, one computer in *group 2*:

```
stars:group2:1:doug $0x4f3c5801
```

One computer to be excluded from dots, two licenses to be reserved for *jill*:

```
dots:group1:2: !$0x592ae8b0 jill
```

In these examples, the two groups denoted as *group1* are distinct because they specify different feature names.

Appendix A - Using Environment Variables to Configure a License Server

This appendix contains information on configuring the license server through the use of the following environment variables:

Table A-2: Run-time Environment Variables

Variable	Where documented
LSDEFAULTDIR	See “LSDEFAULTDIR - Set the Default Location of the License Server Files” on page 129.
LSERVOPTS	See “LSERVOPTS - Set License Server Options” on page 126.
LSERVRC	See “LSERVRC - Set the License Code File” on page 130.
LSERVRCNF	See “LSERVRCNF - Set the License Server Configuration File” on page 130.
LSFORCEHOST	“LSFORCEHOST - Select a Single License Server” on page 125.
LSHOST	See “LSHOST - Set the License Server Name” on page 125.
LS_MAX_QWAIT_SEC	See “LS_MAX_QWAIT_SEC - Set Maximum Queue Wait Time” on page 132.
LS_MAX_QHOLD_SEC	See “LS_MAX_QHOLD_SEC - Set Maximum Hold Time” on page 132.
LS_MAX_QLEN	See “LS_MAX_QLEN - Set Maximum Queue Length” on page 133.
LS_MAX_GROUP_LEN	See “LS_MAX_GROUP_LEN - Set Maximum Group Queue Length” on page 133.
LSPORT	See “LSPORT - Set the License Server TCP/IP Port” on page 131.
LSPROTOCOL	See “LSPROTOCOL - Set Communication Protocol” on page 126.
LSRESERV	See “LSRESERV - Set the License Server Reservation File” on page 131.

A Sentinel LM license server can be configured using operating system environment variables. There is also an environment variable that can be used to tell a protected application which license server to use. These variables are only used when you have a protected application that is used with network license codes. An application that is used exclusively with stand-alone license codes doesn't require a license server. In this case, these configuration variables have no meaning.

This appendix describes the environment variables that can be used to configure a license server. Only a few of these variables are commonly used. The others are not used in most installations.

Note The method of setting an environment variable varies by operating system. A summary of the method for each supported operating system is included in "Notes on Setting Environment Variables by Operating System" on page 134. The examples below show how to set the variable in a Windows 95/98 environment. Be sure to follow the formats below exactly. For example, do not place a space before or after the equal sign in the SET variable statement. Also, all environment variables must be in all uppercase.

An application which has been built to support stand-alone license codes actually includes most of the capabilities of a license server. These capabilities do not include the ability to authorize license codes for other applications. They do include most of the server capabilities that can be selected using the environment variables. Most stand-alone installations never use these capabilities but they are available for those installations with special requirements. You will find a summary of which license server capabilities may be used with a stand-alone application in "Stand-alone Application Use of Environment Variables" on page 135.

Client - Commonly Used Variables

The following environment variables are commonly used on client computers connected to the network.

LSHOST - Set the License Server Name

The LSHOST environment variable is used on a computer that is running a protected application. It tells the application one or more license server computers to contact.

If this variable is not set then the application will perform a broadcast for any license server that can authorize the application to run. When the variable is set, the application will traverse the list of license servers beginning at the first license server in the list and moving down the list. If none of the specified license servers is found, the application will broadcast to the network, looking for any license servers. Separate license server host names with a : symbol.

Example:

To tell protected applications running on a Windows 95/98 based computer to send license requests to a computer on the same subnetwork called ACCT_SERVER and, if ACCT_SERVER is not found, to FINANCE_SRV, place the following in the Windows 95/98 *autoexec.bat* file.

```
SET LSHOST=ACCT_SERVER:FINANCE_SRV
```

LSFORCEHOST - Select a Single License Server

You can use the LSFORCEHOST environment variable to force the application to look for only one license server computer. If the license server listed in the variable cannot be found, the application stops broadcasting and returns an error. LSFORCEHOST overrides a LSHOST environment variable or *LSHOST/lshost* file, and prevents a network broadcast from being done.

Example:

To tell protected applications running on a Windows 95/98 based computer to send license requests to a computer on the same subnetwork called ACCT_SERVER and no other license server, place the following in the Windows 95/98 *autoexec.bat* file.

```
SET LSFORCEHOST=ACCT_SERVER
```

LSPROTOCOL - Set Communication Protocol

The LSPROTOCOL environment variable is used to specify the communications protocol that will be used to communicate with the license server. Specify **IPX** or **UDP** (for TCP/IP) to choose the protocol:

Example:

To tell protected applications running on a Windows 95/98 based computer to send license requests using the IPX protocol, place the following in the Windows 95/98 *autoexec.bat* file.

```
SET LSPROTOCOL=IPX
```

License Server - Commonly Used Variables

The following environment variables are often used on the computer containing the license server to affect license server performance or configuration.

LSERVOPTS - Set License Server Options

The LSERVOPTS environment variable is used to set license server options. All options that can be set using this variable can also be set via command line parameters when the operating system permits. If any command line parameters are present, then the LSERVOPTS variable is ignored by the license server. There are some environments in which it isn't possible to provide command lines parameters. To avoid contradictory settings and to use the same method of

configuring all license servers, it's recommended to always use LSERVOPTS to set these options instead of providing them directly on the command line:

Table A-3: LSERVOPTS Options Summary

Option	Description
-s <i>license-file</i>	Specifies the name and location of the license code file. By default, the license server will use the file, <i>lservrc</i> , in the local directory. The location of the license code file can also be specified by the environment variable, LSERVRC.
-e <i>license-configuration-file</i>	Specifies the name and location of the optional license configuration file, which can contain configuration information for alerts, readable license codes, etc. By default, the license server will use the file, <i>license-file.cnf</i> (appends extension <i>.cnf</i> to the license file path it uses). The location of the configuration file can also be provided by the environment variable, LSERVRC CNF.
-l <i>usage-log-file</i>	Specifies the name and location of the license server's usage log file, which enables usage logging. By default, usage logging is disabled.
-z <i>usage-log-file-size</i>	<p>The maximum size of the usage log file. The size can be specified in bytes, kilobytes, or megabytes. For instance, -z 2000 means 2000 bytes, -z 2k means 2 kilobytes, -z 2m means 2 megabytes. The default value is 1 megabyte. Once the maximum size of the file is reached, the license server will create a backup log file unless the -x option has been used; the -x option causes the license server to simply stop logging transactions.</p> <p>If log file backups are not disabled by -x, when the log file reaches its maximum size the contents of the log file are moved into a new file that has the same name as the original log file but which has two numeric digits appended to its name. For example, if the original log file is named <i>lserv.log</i>, the first backup file will be named <i>lserv.log.00</i>. The next time a backup file is created, the new file will use the next available backup number (e.g., <i>lserv.log.01</i>).</p> <p>The maximum number of backup files is 99; you can move existing backup log files to another directory and the license server will begin again with <i>00</i>.</p> <p>If clients are connected to the license server when the backup file is created, dummy records are created for any pending transactions in the backup file; corresponding dummy records are created in the new file. The dummy records are identified by the user name LM_SERVER.</p>

Table A-3: LSERVOPTS Options Summary (Continued)

Option	Description
-x	By default, on overflow of the usage log file, the file contents will be moved into a backup file. New usage records are then written to the original file until it overflows again. If the -x option is specified, the file will not be backed up on overflow; instead the license server will simply stop writing further records to the file.
-port <i>port-number</i>	The license server port number is used by the TCP/IP protocol when transferring data between the license server and the client. You will rarely need to change this. See “LSPORT - Set the License Server TCP/IP Port” on page 131.
-com <i>percentage</i>	Commuter licensing uses the same license tokens as other network licenses. To ensure that not all license tokens are used up by commuters, set the -com option to the percentage of license tokens you want to be used for commuter licensing. Once that percentage of tokens are used by commuter licenses, no more will be made available to commuters until tokens are returned.
-rlf <i>redundant-license-file</i>	Starts the license server as a redundant license server using the specified redundant license file. You do not need to use this option to start the license server as a redundant license server if a <i>lservrlf</i> redundant license file is in the same directory as the license server.
-lfe <i>encryption-level</i>	Specifies the level of encryption with which license transactions will be written to the license server log file, 1 to 4. For a discussion of why you might want to encrypt transactions and for information on the different levels, see “Encrypting License Server Log File Entries” on page 26. Note that your vendor can override your encryption level choice for specific license codes.
-f <i>error-file</i>	Specifies the name and location of the error file where the license server will log occurrences of unexpected conditions. By default, this is disabled until the option is specified. Then the license server will append to the <i>lservlog</i> file in the current directory.
-u <i>group-reservations-file</i>	Specifies the name and location of the optional group reservations file. By default, the license server uses the <i>lsreserv</i> file in the current directory. The location of the reservations file can also be provided by the environment variable, LSRESERV.
-b	For Windows 95/98, starts the license server as a background service which will not be terminated when the computer user logs off. The license server runs without displaying a window or message. You must use the lsrvdown command to stop the license server.

Table A-3: LSERVOPTS Options Summary (Continued)

Option	Description
-q	Quiet mode. When this option is specified, the license server will start up quietly without displaying its banner. Unexpected conditions will still be logged as usual. (UNIX only.)

Some license server options have a specific environment variable which can be used to set them. You can also set any of these options using LSERVOPTS but this isn't recommended. Any settings made with LSERVOPTS override any settings made using a specific environment variable. To avoid contradictory settings, it's recommended that the specific environment variables be used whenever possible.

Example:

To tell the license server running on a Windows 95/98 based computer to set a 2 megabyte limit on the log file, to stop logging when the file size limit is reached, and to start the license server in quiet mode, place the following in the Windows 95/98 *autoexec.bat* file.

```
SET LSERVOPTS=-z 2m -x -q
```

License Server - Infrequently Used Variables

LSDEFAULTDIR - Set the Default Location of the License Server Files

The LSDEFAULTDIR environment variable can be used to set the default location of the license server files. It's recommended that the license server default directory not be changed. By default, the default directory is set to the directory the license server executable is located in.

Example:

To tell the license server running on a Windows 95/98 based computer that the default directory for license server files is a directory named, *C:\LS_FILES*, place the following in the Windows 95/98 *autoexec.bat* file:

```
SET LSDEFAULTDIR=C:\LS_FILES
```

LSERVRC - Set the License Code File

The LSERVRC environment variable can be used to set the name and location of the license code file. It's recommended that the default name and location are used for this file. By default, this file will be called *lservc* and reside in the license server default directory.

Examples:

To tell the license server running on a Windows 95/98 based computer that license code file is in the current directory but named, *network.lic*, place the following in the Windows 95/98 *autoexec.bat* file.

```
SET LSERVRC=NETWORK.LIC
```

To tell the license server running on a Windows 95/98 based computer that license code file name and path is, *C:\LS_FILES\NETWORK.LIC*, place the following in the Windows 95/98 *autoexec.bat* file.

```
SET LSERVRC=C:\LS_FILES\NETWORK.LIC
```

LSERVCCNF - Set the License Server Configuration File

The LSERVCCNF environment variable can be used to set the name and location of the license server configuration file. This file is used in setting up user alerts and other options. It's recommended that the default name and location are used for this file. In most installations, this file will be called *lservc.cnf* and reside in the license server default directory. If LSERVCCNF isn't used to specify the configuration file then the name and location of this file will be based upon the name and location of the license code file. In this case the configuration

file will reside in the same directory as the license code file and have the same base name as the license code file but with the extension .cnf.

Example:

To tell the license server running on a Windows 95/98 based computer that configuration file is in the default directory but named, *network.cnf*, place the following in the Windows 95/98 *autoexec.bat* file.

```
SET LSERVRCCNF=NETWORK.CNF
```

LSRESERV - Set the License Server Reservation File

The LSRESERV environment variable can be used to set the name and location of the license server reservation file. This file is used by a system administrator to specify the computers and users are authorized to run a protected application. It's recommended that the default name and location be used for this file. By default, this file will be called *lsreserv* and reside in the license server default directory.

Example:

To tell the license server running on a Windows 95/98 based computer that the license code file is in the current directory but named, *reserve*, place the following in the Windows 95/98 *autoexec.bat* file.

```
SET LSRESERV=RESERVE
```

LSPORT - Set the License Server TCP/IP Port

The license server port number is used by the TCP/IP protocol when transferring data between the license server and the client. Do not change the port number unless your vendor has instructed you to do so or conflicts are occurring between the license server and another network application that is using the SentinelLM license server's port number. (The license server default port number is 5093, and has been registered to Rainbow Technologies, Inc.) A symptom of such a conflict would be license server communication errors.

Example:

To tell the license server running on a Windows 95/98 based computer to use the TCP/IP port number 5093, place the following in the Windows 95/98 *autoexec.bat* file.

```
SET LSPORT=5093
```

Environment Variables for Administering License Queuing

Several environment variables allow the network administrator to set license queuing preferences. These environment variables override settings within the application itself originally set by your vendor.

LS_MAX_QWAIT_SEC - Set Maximum Queue Wait Time

This environment variable allows the network administrator to set the maximum queue wait time for the license server computer on which the variable is set.

Example:

To tell the license server running on a Windows 95/98 based computer to set the maximum queue wait time to 15 minutes (900 seconds), place the following in the Windows 95/98 *autoexec.bat* file.

```
SET LS_MAX_QWAIT_SEC=900
```

LS_MAX_QHOLD_SEC - Set Maximum Hold Time

This environment variable allows the network administrator to set the maximum time that the license server will hold a license for a requestor once the license becomes available.

Example:

To tell the license server running on a Windows 95/98 based computer to hold a license for 120 seconds for a user, place the following in the Windows 95/98 *autoexec.bat* file.

```
SET LS_MAX_QHOLD_SEC=120
```

LS_MAX_QLEN - Set Maximum Queue Length

This environment variable allows the network administrator to set the maximum queue length for the license server computer on which the variable is set. It can also override queuing preferences set in the application. Here are its possible values:

- If not set, then the application dictates whether the user is placed in a queue.
- Great than zero: specifies the maximum number of users that can be in the queue.
- Set to less than zero: enables license queuing, even if the application's default is *not* to place the user in a queue.
- Set to zero: disables license queuing for this license server.

Example:

To tell the license server running on a Windows 95/98 based computer to stop placing users in the queue when the queue has 15 users, place the following in the Windows 95/98 *autoexec.bat* file.

```
SET LS_MAX_QLEN=15
```

LS_MAX_GROUP_LEN - Set Maximum Group Queue Length

This environment variable allows the network administrator to set the maximum queue length for the reservation group on the license server computer on which the variable is set. It can also override queuing preferences set in the application. Here are its possible values:

- If not set, then the application dictates whether the user is placed in the group queue.
- Great than zero: specifies the maximum number of users that can be in the group queue.
- Set to less than zero: enables license queuing, even if the application's default is *not* to place the user in a group queue.
- Set to zero: disables group license queuing for this license server.

Example:

To tell the license server running on a Windows 95/98 based computer to stop placing users in the group queue when the queue has 15 users, place the following in the Windows 95/98 *autoexec.bat* file.

```
SET LS_MAX_GROUP_LEN=15
```

Notes on Setting Environment Variables by Operating System

The method used to set an environment variable varies depending upon the operating system. The following is a summary of the methods used for each operating system.

Windows 95/98

Environment variables are set in Windows 95/98 by using the SET statement inside the *autoexec.bat* file.

Windows NT and Windows 2000

In Windows NT/2000, environment variables are set using the **Control Panel | System | Environment** dialog box. Administrator privileges are needed to change these settings. Each environment variable should be added as a System Variable. On Windows NT/2000, environment variables are read by the System

Services when the computer boots. If you make changes to variables, you may need to reboot for them to take effect.

UNIX - Bourne or Korn Shell

On UNIX computers using the Bourne or Korn shell, environment variables are set by using the SET command at the command prompt, in a format similar to that for Windows 95/98. However, remember to enclose the environment variable parameters in quotation marks, and follow the SET command with an export command.

For example:

To set the default license code file name to *net2lic*, type:

```
SET LSERVERC="net2lic"  
  
;export LSERVERC
```

UNIX - C or TCSH Shell

On UNIX computers using a C or TCSH shell, environment variables are set by using the SETENV command at the command prompt in this format:

```
SETENV <variable> parameters
```

For example:

To set the default license code file name to *net2lic*, type:

```
SETENV LSERVERC net2lic
```

Stand-alone Application Use of Environment Variables

When running in stand-alone mode, the license serve code that is part of the application can be configured using most of the environment variables. These

variables are used in the same way as they would be with a network license server. Those variables which are only appropriate for a network configuration are ignored by the stand-alone application. The differences between the use of these variables in stand-alone and network mode is described below.

LSHOST - Disable a Network License

Your application may have been built to support both stand-alone and network licensing. If it includes this capability then it will first try to obtain a stand-alone license using the license server capability that is built into the application. If unsuccessful then the application will try to obtain a network license by communicating with a license server located somewhere on the same subnet. You can disable the network broadcast by setting LSHOST to NO_NET. With this setting, the application only looks for a stand-alone license code.

LSERVOPTS - Set Application License Management Options

This variable can be set to configure the application license management options in a way that is similar to its use on a server computer to configure a license server. It's recommended that the **-z** and **-x** options be used to control the size of the usage file if logging has been enabled. The **-u** and **-q** options aren't meaningful in a stand-alone configuration and shouldn't be used. The remaining options are better set using a specific environment variable.

LSDEFAULTDIR - Set the Default Location of License Management Files

This variable can be used to set the default location of the license code and log files.

LSERVRC - Set the License Code File

This variable can be used to specify the file that is used to store the stand-alone license codes used by the application.

LSERVRCNF - Set the Application License Management Configuration File

This variable can be used to specify the configuration file that is used with the stand-alone application.

Environment Variables Not Used in Stand-alone Mode

The following variables are only meaningful when used with a license server: LSPORT, LSPROTOCOL, and the license queuing environment variables.

Index

A

- additive licenses 16
- alerts 109
 - notification 114
 - setting 109

B

- background service 128
- borrowing threshold 46

C

- client 15
 - configuring 32
- communication protocols 126, 128, 131
- commuter licensing 44, 69
 - checking out authorization 70
 - locking authorization 70
 - redundant license 44
 - redundant license server restriction 70
 - restricting tokens 128
- Computer ID key 5
- conventions
 - syntax xiv
 - typographic xiii
- CSV file 93

D

- distribution criteria 11

E

- echoid utility 79
- encrypting usage log file 26, 128
- environment variables
 - LS_MAX_GROUP_LEN 133
 - LS_MAX_QHOLD_SEC 132
 - LS_MAX_QLEN 133

- LS_MAX_QWAIT_SEC 132
- LSDEFAULTDIR 129
- LSERVOPTS 19, 127
- LSERVRC 82, 110, 130
- LSERVRC CNF 82, 110, 127, 130
- LSFORCEHOST 35, 125
- LSHOST 34, 35, 125
- LSPORT 131
- LSPROTOCOL 126
- LSRESERV 116, 128, 131
- MAILPGM 114
 - stand-alone computers 135
- error logging 27, 128
- event logging, Windows 31
- exclusive licenses 16

F

- files
 - lservlog 26, 128
 - lservrc 127
 - lservrlf 40
 - lshost 34
 - lsreserv 128, 131
- fingerprint 2
 - computing 79
- free pool 41
 - clearing 42

G

- group reservations 115, 128, 131
 - including/excluding computers 116
 - including/excluding users 115
 - reservation file 121
 - setting 115
 - specifying 116

H

hard limits 109

I

including/excluding computers and users 116

IPX protocol 80

ipxecho utility 80

L

-lfe option 26

license balancing 11, 45

 disabling/enabling 45

 free pool 41

 initial setup 49

license code 2

 adding/deleting 84

 decrypting 82

 displaying 81

 installing 84

 loading 18

license file 2, 10, 15, 23, 40, 127

 location 23

 lservrc 23

 lservrlf 23

 setting location 130

license queuing 132–134

license server 3

 -b option 128

 broadcast method identification 34

 client 32

 -com option 128

 commuter tokens, restricting 128

 configuration file 127, 130

 configuring 18, 22

 display information 96

 display list 95

 -e option 127

 error log 128

 -f option 128

 group reservations 128

 identifying 33, 125

 installing 30, 81

 -l option 127

 -lfe option 26, 128

 license file 127

 log file backup 127

 log file encryption 128

 log file location 127

 log file reports 26, 92

 log file size 127

 monitoring 87

 named license server identification 34

 network address, IPX 80

 platform compatibility 17

 port 128

 -port option 128

 -q option 129

 quiet mode 129

 -rlf option 128

 -s option 127

 setting encryption levels 26

 setting location 129

 setting name 125

 setting options 127

 shutting down 89

 starting 28, 30, 82, 83

 starting in background 128

 starting up redundant license server 128

 -u option 128

 user privileges 28

 Windows options 83

 -x option 127

 -z option 127

licenses

 adding and deleting 84

 additive 16

 authorization process 18

- exclusive 16
- renewing 21
- token 38
- loadls utility 30, 81
- locking code 3
- locking commuter authorization 70
- log file encryption 128
- log file. See usage log file
- logging events, Windows 31
- LS_MAX_GROUP_LEN 133
- LS_MAX_QHOLD_SEC 132
- LS_MAX_QLEN 133
- LS_MAX_QWAIT_SEC 132
- lsdecode utility 81, 82
- LSDEFAULTDIR 129
- lserv utility 83
- lserv.log 26
- lserv9x.exe 31
- lservc file 130
- lservlog file 128
- LSERVOPTS 19, 127
- LSERVRC 82, 110, 130
- lservrc file 10, 127
- lservrc.cnf 110
- LSERVRC CNF 82, 110, 127, 130
- lservrlf file 11, 23, 40
- LSFORCEHOST 11, 35, 125
- LSHOST 34, 35, 125
- lshost file 34
- lslic utility 10, 11, 84
- lsmon utility 35, 87
- lspool utility 49
- LSPORT 131
- LSPROTOCOL 126
- LSRESERV 116, 128, 131
- lsreserv file 116, 128, 131
- lsrvdown utility 89
- lsusage utility 23, 89

lswhere utility 34, 95

M

- MAILPGM 114
- majority rule 41
- multi-byte characters 26, 92

N

- NetWare 32
- NetWare license server 32, 80
- network license 3

P

- pool. See redundant license server pool

R

- redundant license 10
 - characteristics 50
 - free pool 41
- redundant license file 11, 40
 - creating 48
- redundant license server 11, 15, 23, 37–45
 - choosing 47
 - failure 42
 - leader 40
 - locking code 48
 - majority rule 41
 - pool 15
 - reservation file 45
 - startup 49, 128
 - user privileges required 43
- redundant license server pool 37, 40
 - maintaining 49
- replacement licenses 16
- reservation file 121
 - configuring 117
 - creating 117
 - group 116
 - members 116

reserving licenses for users 45
rftool utility 11, 40

S

soft limits 109
stand-alone computer 135
stand-alone license 3
syntax conventions xiv
system requirements
 NetWare 32
 Windows 30
 Windows 95/98 30

T

threshold 46
token 11, 38
 distribution criteria 11
token borrowing 45
 enabling/disabling 46
 threshold 46
typographic conventions xiii

U

usage log file 23, 89
 backup 127
 character restrictions 26, 92
 CSV output 93
 encrypting 128
 encryption levels 26
 file name 26
 location 127
 lserv.log 26
 report 89
 reports 26, 92, 93
 setting maximum size 127
user alerts 109
user privileges 28

W

warning messages 109

Windows 30
 event logging 31
 installing license server 30, 81
WlmAdmin utility 34, 96
WlsGrMgr utility 117
Wrlftool utility 11, 40, 46