Sentinel LM Developer's Guide

RAINBOW®
TECHNOLOGIES

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.

Sentinel LM is a trademark of Rainbow Technologies, Inc. Microsoft Windows, Microsoft Windows NT, Windows 95, Windows 98, Windows ME, Windows 2000, Windows 2003 and Windows XP are either trademarks or registered trademarks of Microsoft Corporation in the United States and other countries. All other product names referenced herein are trademarks or registered trademarks of their respective manufacturers.

CONFIDENTIAL INFORMATION

The **Sentinel LM** protection system is designed to protect your applications from unauthorized use. The less information that unauthorized people have regarding your security system, the greater your protection. It is in your best interest to protect the information herein from access by unauthorized individuals.

Part Number 007-0070-001, Revision A
Software versions 7.3.0 and later

Revision	Action/Change	Date
A	Initial Release	January 2004

Rainbow Technologies Sales Offices

United States: *http://www.rainbow.com,* Tel: +1 949 450 7300 / Toll Free: +1 800 852 8569

Australia
Tel: +61 3 9882 8322
http://www.rainbowaustralia.com.au

Brazil
Tel: +55 11 6121 6455
http://www.rainbow.com/brasil

China
Tel: +86 10 8851 9191
http://www.isecurity.com.cn

France
Tel: +33 1 41 43 29 00
http://www.fr.rainbow.com

Germany
Tel: +49 89 32 17 98 0
http://www.de.rainbow.com

Hong Kong
Tel: +852 3157 7111
http://www.rainbow.com

India
Tel: +91 11 26917538
http://www.rainbowindia.co.in

Japan
Tel: +81 3 5719 2731
http://www.rainbowtech.co.jp

Korea
Tel: +82 31 705 8212
http://www.rainbow.com/korea

Mexico
Tel: +52 55 5575 1441
http://www.rainbow.com/latinamerica

Taiwan
Tel: +886 2 6630 9388
http://www.rainbow.com/taiwan

UK
Tel: +44 (0) 1932 579200
http://www.uk.rainbow.com

International Quality Standard Certification

Rainbow Technologies, Inc. Irvine, CA facility has been issued the ISO 9001 Certification, the globally recognized standard for quality, by Det Norske Veritas as of March 2002. Certificate Number CERT-02982-2000-AQ-HOU-RABR2.

European Community Directive Conformance Statement

This product is in conformity with the protection requirements of EC Council Directive 89/336/EEC. Conformity is declared to the following applicable standards for electro-magnetic compatibility immunity and susceptibility; CISPR22 and IEC801. This product satisfies the CLASS B limits of EN 55022.

Contents

Preface

Thank you for choosing the Sentinel LM™ product to license your software. Sentinel LM can provide quick protection for demo programs or more sophisticated protection for full-featured applications. Read on for all you need to know about using Sentinel LM to protect your applications.

Note: This book contains information on using Sentinel LM to create protected applications that run on stand-alone computers or client computers that are part of a network. A network license server manages and maintains all licenses for protected applications running on computers connected to the network.

What Is The Sentinel LM Product Family?

Sentinel LM offers a variety of options designed to give you exactly the features you need for your business environment.

Sentinel LM

The standard Sentinel LM product satisfies most developers needs by providing an easy way to protect and license applications. You receive:

- Sentinel LM-Shell for Windows 95/98/ME and Windows NT 4.0/ Windows 2000/Windows XP/Windows 2003 which protects your 32-bit Windows application by enclosing it in a protective wrapper that will check for a valid license before allowing it to run.

  Sentinel LM-Shell optionally calls the Client Activator program that installs a license code at your customer's site from the Internet or after your customer enters the license code on the keyboard. (The customer may type the license code after obtaining it from a telephone conversation with your fulfillment center or from printed information included with your application packaging.)

- License code generator for Windows 95/98/ME and NT/2000/XP/ 2003, which allows you to create license codes that will be installed on your customer's site to allow use of your protected application.
- Upgrade license code generator for Windows 95/98/ME and NT/ 2000/XP/2003, which allows you to generate upgrade license codes for your protected application.
- License meter key which attaches to the parallel or USB port of an IBM-compatible PC. This key permits use of the license code generator, and allows you to create license codes for applications that will run on a stand-alone or networked computer.
- System administration utilities.
- Online documents in Adobe Acrobat format. (Optional printed manuals are available separately. Contact your Rainbow sales representative for information.)

- Software “license servers” that your customer installs on a Windows or UNIX network computer to manage licenses for applications running on computers connected to that network.
- Sentinel System Drivers for Windows 95/98/ME and NT/2000/XP/2003.

API Option

Although the standard Sentinel LM product offers the easiest to use combination of license management features, if you need more advanced functionality (such as separate licensing of multiple features within a single executable), you will want to purchase the API Option, an add-on to Sentinel LM that includes:

- Sentinel LM Application Libraries for Windows or UNIX that allow you to embed calls to the license management system in your application. Advanced libraries also allow you to create your own license code generator interface and license servers.

The Sentinel LM Manuals

The Sentinel LM product includes several manuals/guides, all designed to work in conjunction with each other.

Manual	What's in it?	Who should read it?
Sentinel LM Developer's Guide	All the steps necessary to protect, package, and ship a stand-alone or network application protected with Sentinel LM-Shell or the Sentinel LM Application Library.	Developers using Sentinel LM-Shell or the API option who are responsible for the overall process of protecting and shipping an application for a stand-alone or network computer.
Sentinel LM Programmer's Reference Manual	Description of the Sentinel LM Application Library.	Developers who are using the Sentinel LM Application Library to protect their applications. This manual assumes you are familiar with the C programming language, although other language interfaces are available.
Sentinel LM System Administrator's Online Guide	Information for the end user of your protected application, including use of administrator commands and configuring and using a license server.	End users of your protected application who are responsible for administering the application and end user license management and who are familiar with system administration tasks.
Sentinel LM Start Here Guide	Information to get the developer up and running with Sentinel LM as quickly as possible. It contains installation instructions and a quick tour of Sentinel LM-Shell.	Any Developer using Sentinel LM.

Manual	What's in it?	Who should read it?
Sentinel LM Release Notes	Information on the features added to the current release. Also contains late-breaking information that was not available when the manuals were completed.	Any Developer using Sentinel LM.

Who Should Read This Guide?

This *Sentinel LM Developer's Guide* that you are currently reading focuses on helping developers who are using Sentinel LM-Shell or Sentinel LM Application Library to protect their product.

It is designed and developed for a higher-level manager or architect who needs to decide what licensing models will be used and how to ship the protected product, and also give the instructions to a programmer who needs to implement the code.

Conventions Used in This Document

The following conventions are used throughout this guide:

Convention	Meaning
Select	Use the arrow keys or mouse to select an item on a menu, a field in a window or an item in a list.
Click	Press the primary mouse button once. The primary mouse button is typically the left button, but may be reassigned to the right button.
`Courier`	This font denotes syntax, prompts, and code examples.
Italic Lettering	Words in italic type represent file names and directory names.
Bold lettering	Words in boldface type represent keystrokes, menu items, window names or fields.
⚠	This warning icon flags any potential pitfalls that we think you may need to be careful of.
Note:	Used for highlighting notes related to a specific section.
Tip:	Denotes tips that should be remembered while trying to achieve a specific result.

How to Get the Most from This Guide

This guide gives all the steps for planning the protection of your application, as well as the steps for protecting, packaging, and shipping your protected application to your customers.

Chapter/Appendix	Description
Chapter 1 – Overview	Shows how Sentinel LM is put together. Also contains a checklist for the overall process of protecting and shipping an application.
Chapter 2 – Installation	Explains how to install and configure Sentinel LM.
Chapter 3 – Planning Your Application Licensing	Provides an overview of planning to protect an application and choosing license models. Contains information on special licensing options such as commuter licensing and license queuing.
Chapter 4 – Protecting Your Application with Sentinel LM-Shell	Information on using Sentinel LM-Shell to protect your application without changing source code.
Chapter 5 – Protecting Your Application with the Application Library	Information on protecting your application by using the Sentinel LM Application Library to add API calls to your application source code.
Chapter 6 – License Activation	Information on using the Sentinel LM Client Activator to install licenses at your customer site.
Chapter 7 – Packaging and Shipping Your Application	Gives the steps for packaging and shipping your application after you have protected it.
Chapter 8 – Creating and Upgrading License Codes	Outlines the use of the license code generator to generate new license codes. For details on using the Windows license code generator, WlscGen, see its online help.
Chapter 9 – Using Redundant License Servers and License Balancing	Contains an overview of the steps necessary to implement license balancing and redundant license servers for you and for your customer.

Chapter/Appendix	Description
Chapter 10 – Using Commuter Licensing	Discusses how you and your customers use commuter licensing to allow temporary use of your application on a portable computer away from the network.
Chapter 11 – Using Capacity Licensing	Provides details about using capacity licensing with Sentinel LM.
Chapter 12 – Using Upgrade Licensing	Discusses how to upgrade licenses for version and capacity with Sentinel LM.
Chapter 13 – Utilities	Provides a discussion of the developer utilities available.
Chapter 14 – Product Add-ons	Gives information on additional products that work with Sentinel LM.
Appendix A – Readable License Codes	Information on different formats of license codes.
Appendix B – Sentinel LM-Shell Error Messages	Provides the Sentinel LM-Shell error codes and their descriptions.

Getting Help

To get familiar with Sentinel LM, use Windows Sentinel LM-Shell, discussed in Chapter 4, “Protecting Your Application with Sentinel LM-Shell,” on page 51.

Online Documentation

Help files are available with Sentinel LM-Shell and the Windows license code generator.

We also provide online versions of the Sentinel LM manuals in Adobe Acrobat format.

For Windows computers, refer to the Adobe Acrobat versions of the Sentinel LM manuals installed on your hard disk. Once the Acrobat Reader is installed, you can read these manuals by using the document shortcuts in *Start* > *Programs* > *Rainbow Technologies* >*Sentinel LM 7.x.x* > *English.*

For UNIX computers, refer to the Adobe Acrobat format manuals in the *SentinelLM/man* directory.

To read and print the online versions of the Sentinel LM manuals, you will need Adobe Acrobat Reader version 4.0 or later. We recommend installing Acrobat Reader 5.0 or higher for better results. These versions of Acrobat can be downloaded from http://www.adobe.com or can be installed from the .

For instructions on obtaining and installing the Acrobat Reader, see “Installing the Adobe Acrobat Reader” on page 33.

If you have questions concerning the Sentinel LM product, contact Rainbow Technologies Technical Support using the information given on page xxiv.

Contacting Rainbow Technologies Technical Support

Rainbow Technologies is committed to supporting the Sentinel LM. If you have questions, need additional assistance, or encounter a problem, please contact Technical Support:

Rainbow Technologies Technical Support Contact Information

Rainbow Technologies Customer Connection Center (C^3)	
http://c3.rainbow.com	
America	
Internet	http://www.rainbow.com/support
E-mail	techsupport@rainbow.com
United States:	
Telephone	(800) 959-9954
Fax	(949) 450-7450
Europe	
E-mail	EuTechSupport@rainbow.com
France	
Telephone	0825 341000
Fax	44 (0) 1932 570743
Germany	
Telephone	01803 RAINBOW (7246269)
Fax	089 32179850
United Kingdom	
Telephone	0870 7529200
Fax	44 (0) 1932 570743

Rainbow Technologies Technical Support Contact Information (Continued)

Pacific Rim	
E-mail	techsupportpacrim@rainbow.com
Australia and New Zealand	
Telephone	(61) 3 9882 8322
Fax	(61) 3 9882 0588
China	
Telephone	(86) 10 8851 9191
Fax	(86) 10 6872 7342
India	
Telephone	(91) 11 2691 7538
Fax	(91) 11 2633 1555
Taiwan and Southeast Asia	
Telephone	(886) 2 6630 9388
Fax	(886) 2 6630 6858

Tip: Check the Rainbow Technologies Web site (*www.rainbow.com*) for the most up-to-date information about Sentinel LM, including FAQs and technical notes.

Export Considerations

Rainbow Technologies offers products that are based on encryption technology. The Bureau of Industry and Security (BIS) in the U.S. Department of Commerce administers the export controls on Rainbow's commercial encryption products.

Rules governing exports of encryption can be found in the Export Administration Regulations (EAR), 15 CFR Parts 730-774, which implements the Export Administration Act ("EAA" 50 U.S.C. App. 2401 et seq.).

Important Note: BIS requires that each entity exporting products be familiar with and comply with their obligations described in the Export Administration Regulations. Please note that the regulations are subject to change. We recommend that you obtain your own legal advice when attempting to export any product that uses encryption. In addition, some countries may restrict certain levels of encryption imported into their country. We recommend consulting legal counsel in the appropriate country or the applicable governmental agencies in the particular country.

We Welcome Your Comments

To help us improve future versions of Sentinel LM documentation, we want to know about any corrections, clarifications or further information you would find useful. When you contact us, please include the following information:

- The title and version of the guide you are referring to
- The version of the Sentinel LM software you are using
- Your name, company name, job title, phone number and e-mail address

Send us e-mail at:

techpubs@rainbow.com

Or, you can write us at:

Rainbow Technologies, Inc.
50 Technology Drive
Irvine, CA 92618

Attn: Technical Publications Department

Thank you for your feedback. Keep in mind that these e-mail and mail addresses are only for documentation feedback. If you have a technical question, please contact Rainbow Technical Support using the information given on page xxiv.

Chapter 1

Overview

This chapter contains information on:

- What is license management and how does it work?
- What is the Sentinel LM license management system and what are its major components?
- What does license management mean to your customer?
- What are the basic steps for using Sentinel LM to protect and ship your application?

Why Use Sentinel LM?

Sentinel LM guards your software against unauthorized use by providing reliable and easy-to-use software protection technology. This technology (which is transparent to the end-user) also offers other benefits:

- Sentinel LM brings you the benefits of electronic commerce by enabling electronic software distribution (ESD) and electronic software licensing (ESL). If you use the Client Activator, your customers can install a license directly from the Internet, reducing your fulfillment costs.
- Sentinel LM vastly increases your marketing reach through full featured evaluations and try-and-buy programs. This is easily done

by providing time-limited licensing capability and conversion of evaluations to purchases with a single license code (without requiring new product installation).

- By providing network-wide licensing, Sentinel LM also helps your users to centralize software license management and reduce cost of software management and deployment. Your software products become enterprise ready with end-user license configuration options and the advanced integration of Sentinel LM license management with enterprise management tools.

What Is Licensing?

Although we'll go into much more detail later in this book, this section gives some simple examples of license management and discusses how Sentinel LM implements licensing.

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.

Stand-alone Application Protection

Let's say you want to sell an application to a single customer on a specific computer (called a *stand-alone computer*). License management prevents that customer from distributing unauthorized copies of your application by locking that application to a specific computer.

When Sentinel LM protects your stand-alone application, it embeds the license management function that checks for a valid license code before the application is run. Very simply, if a valid license code exists, the application will run; if Sentinel LM cannot find a valid license code, the application will not run.

You may also restrict the application to run only on a specific computer by locking the application to that computer.

Note: A *license* provides an authorization to use a protected application and is embodied by a license code. A *license code* is a string of encrypted characters that Sentinel LM consults to determine whether or not to grant a license. The license code contains all of the information that defines the license agreement between you and your customer: whether the license will expire after a specific number of days, whether the application can only run on a specific computer, and so on.

Here's an example of stand-alone licensing. You sell a word processing application that contains three different programs:

- Word processing
- Spell checking
- Thesaurus

Customer Abbot purchases a license from you that activates only word processing. Every time this customer starts the word processor, Sentinel LM verifies that a valid license code for that program exists and grants authorization to use the program. If customer Abbot attempts to use spell checking, Sentinel LM denies the request to use the program and tells your customer that a valid license code does not exist for that program.

An important feature of Sentinel LM is that customer Abbot can at any time purchase a new license code from you that activates the other two programs of your application. You send customer Abbot a new license code and no new software needs to be installed.

Customer Brown purchases a more expensive license code from you that activates all three programs of your application. When customer Brown uses each program, Sentinel LM verifies that the installed license code permits use of all three programs and grants authorization for their use.

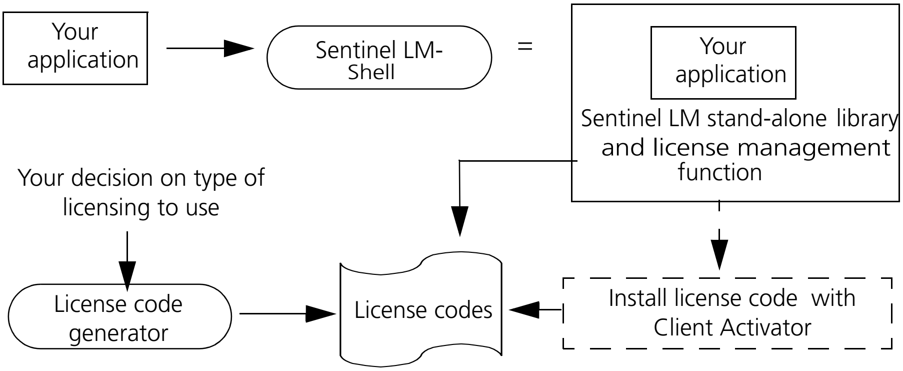
How Does Stand-Alone Licensing Work?

There are two basic methods of protecting an application using Sentinel LM: using Sentinel LM-Shell or using the Sentinel LM Application Library.

When you protect a stand-alone application using Sentinel LM-Shell, it “wraps” your application executable file in a protective wrapper that contains many of the Sentinel LM Application Library calls and license management functions. This allows you to protect an application without modifying the program source code.

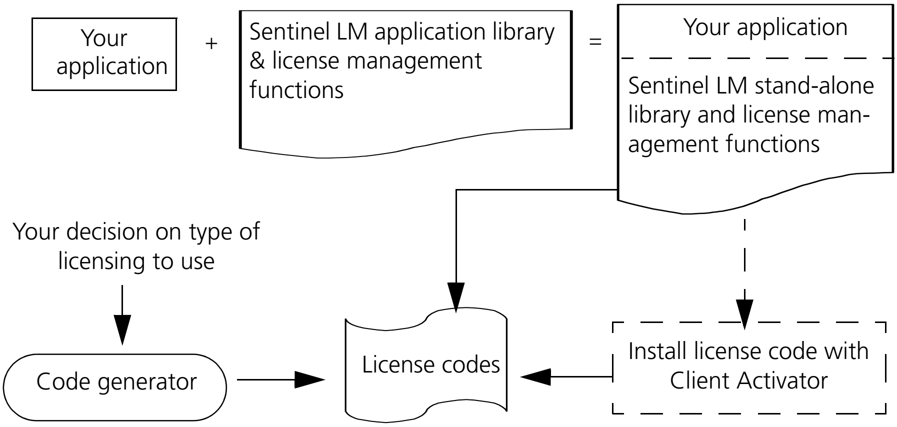
As shown in the illustration below, after protecting your application, you generate license codes that are the physical representation of the license agreement you made with the purchaser of your application. The license code defines, for example, exactly which of your applications can be used by the customer, whether the application can run on any computer or only on a specific one, whether the license will expire on a specific date, and so on.

The license management functions embedded in the application consults the license codes installed on the computer to grant or deny authorization to use the application.


Your application
Sentinel LM-Shell
=
Your application
Sentinel LM stand-alone library and license management function
Your decision on type of licensing to use
License code generator
License codes
Install license code with Client Activator

Protecting a Stand-Alone Application with Sentinel LM-Shell

When you use the Sentinel LM Application Library to protect a stand-alone application, you incorporate library calls in your program source. Various library and license management functions actually become part of your application. Just as with Sentinel LM-Shell protected application, you generate license codes that reflect your license agreement with your customer, and the protected application grants and denies authorization to use the application based on the license codes installed on your customer's computer.


Your application
+
Sentinel LM application library & license management functions
=
Your application
Sentinel LM stand-alone library and license management functions
Your decision on type of licensing to use
Code generator
License codes
Install license code with Client Activator

Protecting a Stand-Alone Application with the Application Library

Installing a License Code for a Stand-alone Application

Whether your application is protected with Sentinel LM-Shell or by Application Library calls, it can optionally call Sentinel LM license activation programs to install your application's license at your customer's site if a valid license is not found.

For stand-alone applications, you can use Sentinel LM Client Activator to make license installation easy and simple for your customer. The Client Activator allows the customer to install a license directly from the Internet or by typing the license code from the keyboard, from a label on the product packaging or when obtained over the telephone by calling your *fulfillment center* (A company that stocks software, manuals, keys, etc. and puts everything together to ship to a customer. Your fullfillment center might also act as a distributor, and program keys for your customer). To use the Client Activator to install licenses over the Internet, you must also have the Sentinel Express product, available separately. The Client Activator gets the license from internet and installsl it on local machine. For more information on license activation and the Client Activator, see Chapter 6, "License Activation," on page 61.

Stand-alone Licensing on a Terminal Server Disabled

To prevent multiple users from logging onto a terminal server to run an application, thus defeating the intention of the application's license to authorize use by a single customer, Sentinel LM automatically detects if a stand-alone application is being run on a terminal server and terminates the application with an error message. An application protected for network mode will be allowed to execute on a terminal server.

Note: On a Windows 2000/ 2003/ XP server with terminal services enabled, a stand-alone application will run if the user is executing the application directly on the server (that is, the user is not remotely connected). On Windows NT terminal servers, the stand-alone application will not execute under any circumstances.

Network Application Protection

In the case where you sell your application to run on multiple computers on a network, Sentinel LM moves the license management function outside of the protected application and uses an external *license server* to verify that a valid license code exists before granting authorization to run the application. (The license server can run on any computer on the network, including a computer on which the protected application is run.)

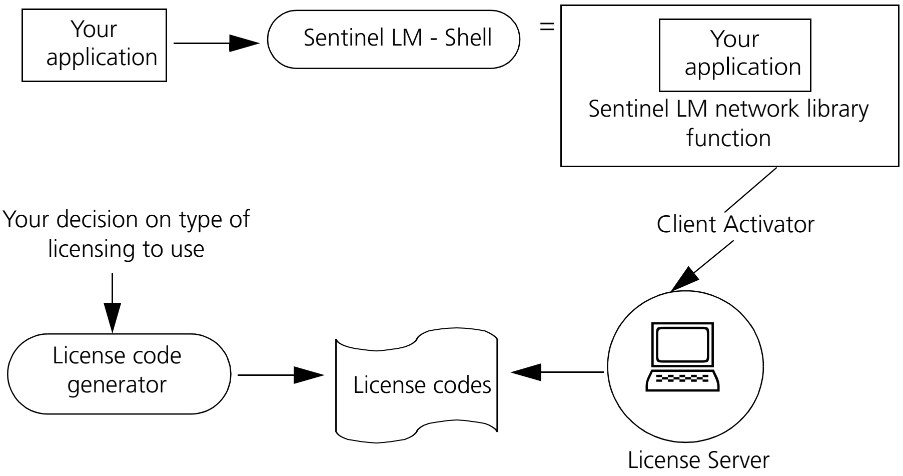
The license server keeps track of all Sentinel LM licenses and handles requests from network users who want to run your application, granting authorization to the requesters to allow them to run the application, and denying requests when all licenses are in use.

Here's an example.

A company purchases a license for your application that allows up to 10 users to use the application at the same time. (We say that the single license has 10 *tokens*.) A user on one computer starts up your application; the application requests a license token from the license server. The request is approved, and the application runs. While that application is running, nine other users on other computers try to run your application. Each request is granted because license tokens are available. However, when the 11th user tries to run the application, the request is denied because no license tokens are free, and the application tells the user that no licenses are available. When one of the existing 10 users exits your application, a license token is freed and the new user can try again to run the application.

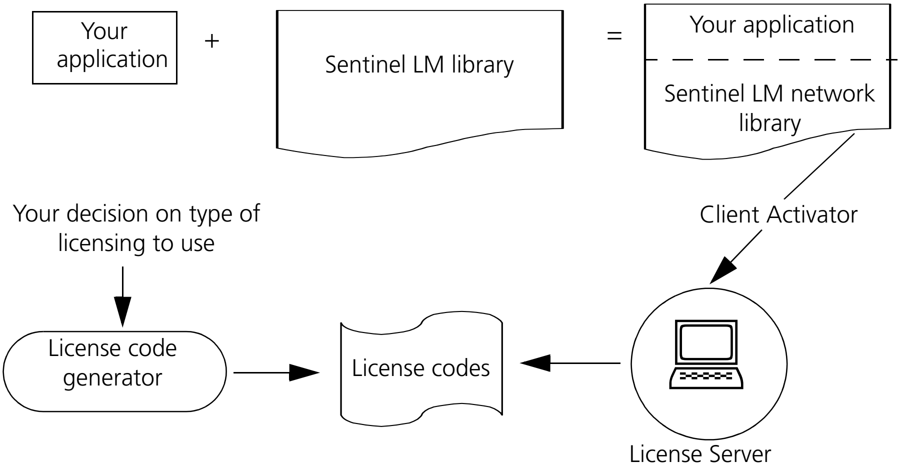
How Does Network Licensing Work?

When you use Sentinel LM-Shell to protect your application and you specify that the application will run in a network environment, Sentinel LM-Shell wraps your application in a protective shell that contains Sentinel LM Application Library functions. In this case, an external license server handles all license management. As with all Sentinel LM-protected applications, you use the license code generator to create the license codes. The license server consults the license codes to determine whether or not to grant a license.



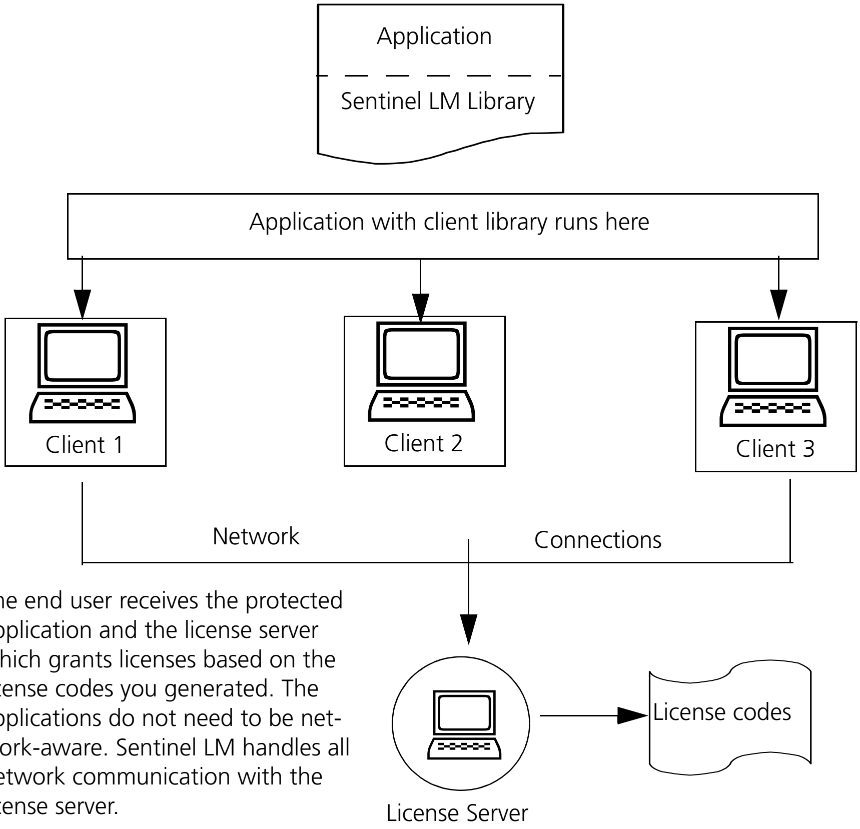
Protecting a Network Application with Sentinel LM-Shell

When you use the Sentinel LM Application Library to protect a networked application, you incorporate library calls in your program source. Again, you generate license codes that are installed on the customer's computer where the license server resides. The license server consults the license code to determine whether or not to grant licenses.



Protecting a Network Application with the Application Library

Typically, your customers install your application on one or more computers or on a file server which is connected to the network. They designate one computer on which the license server will run (the computer need not be the file or application server). To obtain a license, the client applications communicate with the license server over the network as soon as they start up. Only when a valid license code is found does the license server grant authorization to run the application. Applications need not be network-aware. Sentinel LM handles all network communication with the license server.



The end user receives the protected application and the license server which grants licenses based on the license codes you generated. The applications do not need to be network-aware. Sentinel LM handles all network communication with the license server.

What the End User Receives

Since all applications communicate with the license server when they start up, the license server can keep a tally of the number of license tokens in use. The license server then ensures that only the specified number of copies of the application are allowed to run concurrently.

Once successfully started, the application must periodically confirm its continued existence to the license server by renewing the license while the application is still running. (The period of time between renewals is called the license *lifetime*.) This ensures that the license server releases the license and makes it available to other application just in case the application crashes without releasing the license.

License Installation for Networked Applications

Whether your application is protected with Sentinel LM-Shell or by Application Library calls, it can optionally call Sentinel LM Client Activator license activation programs to install your application’s license at your customer’s site if a valid license is not found. For more information on license activation, see Chapter 6, “License Activation,” on page 61.

Integrated Application Protection

Sentinel LM provides an integrated library that allows an application to switch between stand-alone and network licensing. The benefit of using the Sentinel LM integrated library lies in the fact that the license type (stand-alone/ network) can be decided when the application is run, not when the application is compiled.

At the time you protect your application, you may not know whether it will be used in a stand-alone or network environment. You can protect your application for either environment! At the time you generate a specific license code for your customer, you can make the final decision on whether the application will be installed in a stand-alone or network environment. For information on protecting your application for both stand-alone and network use, see Chapter 4, “Protecting Your Application with Sentinel LM-Shell,” on page 51 and Chapter 5, “Protecting Your Application with the Application Library,” on page 59.

Dynamic Switching Between Stand-alone and Network Licensing

One of the most important benefits Sentinel LM offers is *dynamic switching* when an application is protected using the Sentinel LM integrated library and when the LSFORCEHOST environment variable is not set to any value. Dynamic switching occurs when an application is protected with the integrated library and it is not defined at that time whether the application will obtain a stand-alone license on the computer on which it is running (stand-alone mode) or from a license server (network mode). At the time the application is run, the decision is made based on the availability of a license.

The LSHOST environment variable is interpreted as stating a preference for where a license will be obtained:

- If LSHOST on the client computer is not set to anything, the application protected with the integrated library will search first for a stand-alone license on that computer; if it is not found, the application will start looking for network license servers for a network license.
- If LSHOST is set to anything other than NO-NET, the application will look first for a license server on the computers named by LSHOST; if it cannot find a license server on the computers defined by LSHOST that can grant the requested license, Sentinel LM will do a broadcast on the network to look for a network license server that has the license. If the appropriate license server is not found on the network, Sentinel LM will start looking for a stand-alone license.

If LSHOST is set to NO-NET, Sentinel LM will first look for a stand-alone license; if it cannot find one, it will start looking for a license on a network license server.

Note: Multiple license server name can be defined in LSHOST environment varaible separated by **:**

Tip: You can use the LSFORCEHOST environment variable to force Sentinel LM to look for a license on one specific computer. Or to force Sentinel LM to ONLY look for a stand-alone license, set LSFORCEHOST to NO-NET.

Examples of Dynamic Switching

- If LSHOST is set to NO-NET followed by a list of computers containing license servers:

  ```
  LSHOST NO-NET:ACCTNG1:MIS2:ORION
  ```

  If a stand-alone license is found, the application will use it. If a stand-alone license is not found, Sentinel LM switches to searching the computers on the network named in the LSHOST environment in the order listed for a license server containing a network license that can be granted.

Server will also do brodcast in the end if no license is found on the named servers.

- If LSHOST is set to a single computer name, Sentinel LM starts looking for a network license from a license server on that computer. If it doesn't find a license, it performs a broadcast on the network to look for a network license server that has the license. If an appropriate license server is not found on the network, Sentinel LM switches to stand-alone mode and looks for a stand-alone license on the computer on which the application is running.

Overview of Sentinel LM

The following sections detail the components that form Sentinel LM.

Sentinel LM-Shell

Sentinel LM-Shell is an easy-to-use tool for Windows 95/98/ME and NT 4.0/2000/XP/2003 computers that gives you a way to quickly "wrap" your application with a protective shell in just a few minutes. Sentinel LM-Shell is designed to be easy to use, quickly satisfying the most common developer needs. Sentinel LM-Shell is ideal for quickly protecting demonstration programs that use a time-limited license or for protecting an application for which you do not have source code. For details on using Sentinel LM-Shell, see Chapter 4, "Protecting Your Application with Sentinel LM-Shell," on page 51.

Sentinel LM Application Library

You integrate the Sentinel LM Application Library calls into each licensed application to ensure that the application does not run without first obtaining authorization. Use Sentinel LM Application Library for more control over licensing, or if you want to use particularly sophisticated licensing models to modify your program code.

For information on using the Sentinel LM Application Library to protect your application, see Chapter 5, “Protecting Your Application with the Application Library,” on page 59. And for a complete description of Sentinel LM Application Library, see the *Sentinel LM Programmer’s Reference Manual*.

The License Server

The Sentinel LM license server is the central administrator of all licenses for protected applications configured for network use. All licensed applications must receive authorization to execute from a license server. The Sentinel LM license server supports 1000 licenses which includes additive licenses also. A single license server can maintain licenses for all applications licensed by Sentinel LM on the local network.

Included with Sentinel LM are license servers for Windows 95/98/ME, Windows NT/2000/XP/2003, and UNIX.

Servers may exist on different platforms than the clients running the licensed application. For example, a license server running on UNIX may administer Windows 95/98/ME clients. All license server provide backward compatibility to previous version clients.

Licenses

License codes are produced by the Sentinel LM license code generator, which prompts for information about the feature use and the licensing agreement. This code can then be given to the intended customer. Installing the license code causes the Sentinel LM license server or the protected stand-alone application to recognize and enforce the license agreement. (For information on installing licenses, see Chapter 6, “License Activation,” on page 61.)

The following minimal information is contained within all license codes:

- Identification of software vendor
- Feature name

- Computer locking information. (The license can be locked to a particular computer based on its "fingerprint" or can be unlocked to run on any computer.)
- Expiration date or number of days before license expires.
- Number of concurrent users that can use the application.
- Lifetime of each license granted (period of time between license renewals).

Additional information can be stored depending on your requirements. License codes may contain only ASCII characters in the range of 32-127, which excludes multi-byte characters and accented characters. For more information on license codes, see Chapter 8, "Creating and Upgrading License Codes," on page 79.

License File

License codes, when installed, are contained in a *license file*.

For stand-alone applications, the license file containing the license code is by default named *lservrc*, and must be in the same directory as the protected application.

For network applications, the license file (also named *lservrc*) resides in the same directory as the network license server. You can use different file name (other than *lservrc*) through environment variables and command line switches.

Redundant license codes (configured to be serviced by a pool of redundant license servers) are contained in the *redundant license file*, by default named *lservrlf*. License codes that will not be serviced by the redundant license server pool are in an *lservrc* file. A redundant license server will service licenses in both the *lservrc* and the *lservrlf* files, but the licenses in the *lservrc* file will be serviced only by the local license server and not the entire redundant license server pool. For information on redundant license servers and redundant license codes, see Chapter 9, "Using Redundant License Servers and License Balancing," on page 137.

Popular License Models

Many different types of licenses can be defined by a Sentinel LM license code. The type of license model you use depends on your customers' needs and on your application. Details are contained in Chapter 3, "Planning Your Application Licensing," on page 35. For now, here are just a few of the most commonly used license models:

Demo License

Licenses that limit the time your customer can use your application or that limit the use of application features are often shipped with evaluation versions of software. The idea is to let the customer try and then buy your software. When the customer purchases your software, you ship a new, permanent license that is not time-limited or that enables all the features of your application.

A time-limited license allows the user to try the application until the license expires; at that time, the customer must purchase a new, permanent license. The time-limited license is often not locked to a specific computer to allow a customer to run the protected application on any computer.

This type of license is also called a temporary license, since it is not used just for demonstration software, but also can be shipped with full-featured software to allow the customer to get up and running for a short time before he or she needs to obtain a permanent license from you.

You can either set a time-limited license to expire on a specific date or to expire a certain number of days after the application is first run at the customer's site.

Trial License

This type of demo license is a temporary license that starts "ticking" the first time the application is run and will go for a specific number of days. If a 30-day license is issued, but the application is not started and is then run after say 20 days, the license will then start "ticking" and 30 days later (50 altogether) the license will expire.

Sentinel LM uses advanced protection technology to ensure that the customer cannot reuse the license even if the application is uninstalled from the computer, the computer is rebooted, and the application is reinstalled before trying to install the license again.

Client-Locked License

You can lock your protected application to a specific computer to prevent your customer from using the application on more than one computer. The locking is done based on your customer's computer *fingerprint*. You determine what elements of your customer's computer will make up its fingerprint (Ethernet address, disk ID, etc.).

Shared License

A shared license allows a network customer to use multiple functions of the application without consuming more than one license. This is especially useful when a single application contains multiple functions that would normally be used together, such as spell checking and a thesaurus. In addition, suites of programs can be licensed in this manner.

The License Code Generator

The license code generator encrypts the information of a licensing agreement into a license code in the form of a text string. These license codes are then decrypted by Sentinel LM to determine how many simultaneous users are allowed and with what special restrictions. License codes generated can be used on any platform.

Quick and Easy License Codes: Predefined License Types

The most popular types of license codes are predefined in the license code generator and are optimized to produce all-numeric, short license codes. Using a predefined license type makes creating a license code especially quick and easy because many of the license settings are already defined for you.

For a complete discussion of the license code generator, see Chapter 8, "Creating and Upgrading License Codes," on page 79.

Multi-feature License Codes

You can offer your customers the convenience of a single license code that licenses a range of features. The *multi-feature license code* is typically used to license an entire set of applications with a single license code. For details, see Chapter 8, “Creating and Upgrading License Codes,” on page 79.

Upgrade License Code Generator

The upgrade license code generator upgrades the version or capacity or both(version and capacity) of your base license (licenses generated by the license code generator). Sentinel LM allow version upgrade for both capacity and non-capacity (Pooled and Non-pooled) licenses. Use the upgrade license code generator to upgrade both version and capacity of non-pooled licenses.

For more details on Upgrade license code generator see Chapter 8, “Creating and Upgrading License Codes,” on page 79.

Commuter Licenses

Business travelers would like to use a licensed application on a laptop computer when away from the office. Commuter licensing allows authorizations to be checked out from the network onto a laptop computer to allow temporary use of the application, and then to be checked back in when the traveler returns. Commuter licenses are temporary (maximum of 30 days). For more information on commuter licenses, see Chapter 10, “Using Commuter Licensing,” on page 155.

Capacity Licenses

Capacity license of an application defines the number of operations that can be executed per hard limit and allows end user to use an extra flag along with hard limit to control numer of applications that can run simultaneously.

One token of a capacity license allows a set number of operations to be performed (like monitoring 10 door-access sensors). A single token is shared by a team of users at your customer's site which has been assigned that token.

The sharing limit field in the license specifies the team size, which restricts the number of members each team can have.

For more details on capacity licenses refer to Chapter 11, “Using Capacity Licensing,” on page 165.

License Queuing

If license queuing is enabled for an application, when a customer runs that application but no licenses are available, the customer is placed in a queue until a license becomes available. License queuing can be enabled when protecting the application by using Sentinel LM-Shell or by using the Sentinel LM Application Library. For details on enabling license queuing, see Chapter 4, “Protecting Your Application with Sentinel LM-Shell,” on page 51 or the *Sentinel LM Programmer’s Reference Manual*.

License Balancing and Redundant License Servers

One of Sentinel LM’s most powerful features is the use of redundant license servers. Redundant license servers can provide two main benefits:

- *License balancing* distributes license tokens among multiple redundant license servers to reduce the load on a particular license server. Each license server can serve any end user and as time goes by, the license servers redistribute license tokens based on actual license requests, ensuring that the speed with which end users receive tokens is optimized.
- *License server backup.* If one license server goes down, other license servers in the redundant license server pool handle the licenses serviced by the original license server to provide uninterrupted service to the network end users, and end users automatically switch over to using a new license server.

  For information on license balancing and redundant license servers, see Chapter 9, “Using Redundant License Servers and License Balancing,” on page 137.

Because setting up redundant license servers and license balancing takes place at your customer’s site, instructions are included for your customers in the *Sentinel LM System Administrator's Online Guide*. The developer's side of redundant license servers is creating license codes to work with them; see Chapter 8, "Creating and Upgrading License Codes," on page 79.

Client Activator

Sentinel LM includes license activator programs that you can use to install a license if a valid license is not found on your customer's site. For information on automatically installing a license from the Internet or allowing your customer to install a license by entering it on the keyboard, see Chapter 6, "License Activation," on page 61.

Computer ID Keys

The Computer ID key is a hardware device that provides a fingerprint for a specific Windows computer, allowing you to lock the license server or the protected application to it. You can think of a Computer ID key as a portable computer fingerprint.

The Computer ID key attaches to the parallel or USB port on IBM-compatible PCs. Because the Computer ID key passes data through, other connectors can be piggybacked onto the parallel port version of the key, allowing you to continue to use that port for other things (such as a printer).

Computer ID keys are particularly useful when you want to give your customer the flexibility of moving the license server or application to different computers, or if the computer fingerprint is likely to change because hardware components such as Ethernet cards and disk drives are frequently moved between computers. Computer ID keys also provide an extremely secure method of identifying a computer.

There are two situations in which developers lock a license to a Computer ID (CID) key:

- A stand-alone application's license is locked to a CID key on the stand-alone computer;
- An application gets its license from a network license sever and its license is locked to the CID key on the user's computer (client-locked)

or the license is locked to the CID key on the computer on which the license server is running (server-locked) or both.

Cascaded Computer ID Keys for Stand-alone Licenses

You can attach up to five different CID keys to the computer's parallel port, and Sentinel LM will search each for the locking code specified in the stand-alone license.

Note: The utilities that display information from a CID key (such as Wechoid or echoid), will only see the first CID key physically attached to the computer's parallel port.

Cascaded Computer ID Keys for Network Licenses

If a license is server-locked: Up to five different CID keys can be attached to the parallel port of the computer on which the license server runs, and applications requesting the licenses locked to any of those CID keys can run concurrently on the network.

If a license is client-locked (or server-locked AND client-locked): Sentinel LM will only look at the first CID key (the one physically next to the computer) on the client computer for the client locking code.

Sentinel LM Character Sets Supported

Sentinel LM does not support unicode character sets, but does support multi-byte characters in some cases.

Any text that is stored within a license code (for example, feature name, feature version, challenge-response data, etc.) must be entered as 7-bit ASCII form (that is, using the range of ASCII characters 32-127, which excludes multi-byte characters or accented characters). Additional items that must also be entered as 7-bit ASCII characers are:

- Publisher name
- Redundant license server pool names
- WlscGen log comments

Other data, such as user names and host names may be entered as multi-byte characters. Sentinel LM assumes the correct code page for the computer has been enabled if you are going to use multi-byte characters.

What Does Licensing Mean to Your Customer?

What does licensing mean to the customer who purchases your protected application? To a great extent, application licensing is invisible to your customer. Although a network or system administrator can maintain and change Sentinel LM configuration, the end users of the application may never realize license management is in place unless a license expires, in which case they will receive a message asking for a new license code.

Here is a general idea of how license management is implemented at your customer's site for a specific computer:

1. Your protected application is installed on your customer's computer, which may be a stand-alone computer or a networked computer. (If your application is to run on a network, the customer also will install and configure the license server.)

2. If you have chosen a license model that does not require knowledge of the customer's computer (example, time-limited demonstration license), you can ship a license code with the application. In this case, the customer can go to Step #4, below.

   If the license model you have chosen requires locking the application to a specific computer or locking the license server to a particular computer, the customer uses the **echoid** program to compute the fingerprint of the computer to be locked. (If you are using the Client Activator to install a license, the customer does not need to use echoid because the Client Activator automatically computes and displays the fingerprint.) The customer sends the fingerprint to you, the developer of the application

3. You, the developer, will use the WlscGen or lscgen program to generate a license code based on the computer fingerprint and return

that license code (an alphanumeric or numeric string) to the customer.

4. The customer installs the license code. If your application has been developed to accept a license code at installation time, all the customer has to do is enter the license code. If the application has not been developed in this way, the customer will use a license installation program such as lslic to install the license code or license code file.

Customers can also enable various end user options, such as excluding specific computers or users from running the protected application.

For details on commands and features available to the end user of your application, refer to the *Sentinel LM System Administrator's Online Guide*.

End user utilities are available to allow a system administrator to control how licenses are to be used. For a list of Sentinel LM commands you can distribute to your customers, Chapter 13, "Utilities," on page 193.

Steps to Protect and Ship Your Application

The rest of this book gives detailed information on how to protect, package, and ship your application. Here is a brief checklist of the basic steps.

Step No.	Description
#1	**Install Sentinel LM.** Install Sentinel LM on your development computer. See Chapter 2, "Installation," on page 25.
#2	**Use the Tutorials.** *Sentinel LM Start Here Guide* will guide you through licensing an application. (For UNIX, refer to the *Sentinel LM Programmer's Reference Manual*.)
#3	**Plan Your Licenses.** See Chapter 3, "Planning Your Application Licensing," on page 35.
#4	**Protect Your Application.** Protect your application by using Sentinel LM-Shell or Sentinel LM Application Library. See Chapter 4, "Protecting Your Application with Sentinel LM-Shell," on page 51 and Chapter 5, "Protecting Your Application with the Application Library," on page 59.
#5	**Plan License Activation.** Choose how your customer will install a license if no license is found at the customer's site. See Chapter 6, "License Activation," on page 61.
#6	**Package and Ship Your Application.** See Chapter 7, "Packaging and Shipping Your Application," on page 67.
#7	**Create License Codes.** Use the license code generator to create license codes for the end user of your application. See Chapter 8, "Creating and Upgrading License Codes," on page 79.

Chapter 2

Installation

Note: This chapter is **Step #1, "Install Sentinel LM,"** of protecting and packaging your application. For a list of these steps, see Chapter 1, "Steps to Protect and Ship Your Application," on page 23.

This chapter provides installation instructions and other information for:

- Windows
- UNIX

Before You Begin

There are several settings you need to check on the computer you are going to use for your Sentinel LM development. These are discussed below.

Check the Computer Time and Date

It is important to check that the computer time and date are set correctly before installation because many Sentinel LM licensing models check for tampering with the computer's date and time. Users of time-limited licenses must not change the date and time after licenses have been installed on the computer or authorization for such licenses may fail.

Note: This caution also applies to your customers. If your customer changes the date on the end user computer and is no longer able to use your license, you may want to send the customer a one-day license and a utility to clean up the computer. For information on resetting a computer on which time tampering or re-use of a trial license has been detected, see "Integrated Cleaning Tool (lsclean and lscgcln)" on page 202.

Check Communication Protocols

Using Sentinel LM to protect applications to be used in a network environment requires that the computer on which you are installing Sentinel LM use the Windows TCP/IP or NetWare IPX/SPX (Microsoft) network protocol.

If you are developing stand-alone applications that do not need to run in a networked environment, you do not need a communication protocol or a license server installed.

To verify that TCP/IP and/or IPX/SPX are installed on your Windows computer:

1. From the **Control Panel**, double-click the **Network** icon.
2. On the **Configuration** or **Protocol** tab, look at the list of Network components installed to see if TCP/IP or IPX/SPX are listed.

Note: This caution aplies to your customers. It is important to check the communication protocols while installing the protected application.

Windows 95/98/ME and NT/2000/XP/2003

The installation process described is for Windows 95/98/ME and Windows NT 4.0/Windows 2000/XP/2003. Any differences are noted.

On Windows NT/2000/XP/2003, in order to set up Sentinel LM correctly and to install the license server, you *must* have Administrator privileges.

Computer Requirements

Sentinel LM modules run on PCs using a 80386 processor or higher. The standard package supports both TCP/IP and NetWare IPX/SPX protocols.

If You Are Using NetWare

IPX clients can be served by a Win32 server (*lserv9x.exe* or *lservnt.exe*) running on Windows 95/98/ME or Windows NT/2000/XP/2003 computers. This does not require an NLM or running any application on the NetWare server.

Note: If you wish to use IPX protocol for client-server communication then you need to set the LSPROTOCOL environment variable to IPX at the client end.

Installing the Software on Windows

Note: All required libraries and binaries will be encoded during installation with your unique customer information, called a *developer ID.* The developer ID serial number is registered with Rainbow Customer support and must be specified when support for additional platforms is purchased.

To install:

1. Insert the CD in the appropriate drive.
2. Using Windows Explorer, access the CD and double-click on *setup.exe* and double-click the file name. Setup will start up automatically if CD autorun is enabled. The Sentinel LM software installer is displayed.

   You can install the **Sentinel LM Software**, **Sentinel Client Activator 7.0**, **Sentinel System Driver**, and **Sentinel LM Server.**
3. To install Sentinel LM software click **Install** for **Sentinel LM 7.x.x. Software**.
4. The InstallSheild Wizard for Sentinel LM guides you through Sentinel LM installation process. Click **Next** to proceed further.

5. You are asked to select a language. The computer on which you install Sentinel LM must be configured to display and use the language you select. Click **Next** to continue.

6. Click **Yes** to accept the license agreement.

7. Now you are asked to enter the customer information. Type your name and company name. Click **Next** when finished.

8. Sentinel LM asks you to enter a serial number. This serial number is contained in Sentinel LM product packaging and is unique for each developer. Entering this number customizes your version of Sentinel LM. If you do not enter a serial number, Sentinel LM will be installed as evaluation software. Your serial number also tells Sentinel LM your developer ID, if you are installing the standard product, and if you have purchased the API Option.

   After you enter your serial number, click **Next**.

9. In the **Choose Destination Location** dialog box, either accept the default location of *Program Files\Rainbow Technologies\SentinelLM 7.x.x\English* or choose a different location.

   To choose a destination other than the default, click the **Browse** button. A **Choose Directory** box will appear. Navigate through the drives and directories until you arrive at the location you want, then click **OK**.

   When you have chosen your destination, click **Next**.

10. The **Select Program Folder** box is displayed.You may use the default folder name, *Rainbow Technologies*, or you can use an existing folder or create a new one.

    When you have selected a program folder, click **Next**.

    The setup program now copies the files and creates icons in the program folder. When it is done, the **Setup Complete** window displays.

11. Sentinel LM has been installed. **Rainbow Technologies > Sentinel LM 7.x.x > English** will be added to the list of programs accessible through the **Start** menu.

12. Also install the **Sentinel Client Activator 7.0** by clicking Install on the Sentinel LM installer.

13. When setup is complete, click **Finish**.

Note: Many of the Sentinel LM files are customized with your specific developer ID during installation. However, the *Demo* directory and subdirectories contain files that have been customized with a demo ID. It is very important that you not move the demo client library and demo license file in the *Demo* directory or its subdirectories to other Sentinel LM directories, since this would overwrite your customized files, resulting in your protected applications not being able to acquire licenses.

Installed Directories

Sentinel LM is now installed.

Note: If you have installed Sentinel LM without a license code, an evaluation version of the software has been installed. You will still see many of the directories and files listed below.

The *\Program Files\Rainbow Technologies\SentinelLM\7.x.x\English* directory contains the following files and subdirectories:

The directory (*SentinelLM\7.x.x\English*) contains a *ReleaseNotes.pdf* file. That directory also contains uninstallation files.

When Installed	Directory	Subdirectory	Contents
All installations.	*Delivery*	*Admin.net* *Admin.sta*	Administrator utilities for network and stand-alone licensing.
		Driver	Sentinel System Drivers for different Windows platforms.

When Installed	Directory	Subdirectory	Contents
		Server	License servers for Windows 95/98/ME and Windows NT/ 2000/XP/2003. Also contains installation program and e-mail alert utility.
		Your_App	Future location of your protected application.
All installations.	*Demo*	• Client • Licenses • Msvcdev • Server	Sentinel LM demo program and files used by it. Also sample files and libraries.
All installations.	*LM-Shell*		Sentinel LM-Shell.
All installations	*Manuals*		Online manuals in Adobe Acrobat format. See the *ReadMeFirst.pdf* document for information on where all the information on Sentinel LM is located. Programmers Reference Manual will only be included in developer's installation.
Installation for Developers	*Msvcdev*	*Custom*	Files to customize Sentinel LM components.
		Include	The include files, which provide the prototypes for various Sentinel LM library functions.
		Lib	The client libraries.
		Samples	Examples that illustrate the various features of Sentinel LM.
		Your_Src	Future location of your application source files.
Installation for Developers	*Interface*		Programming interface samples.

When Installed	Directory	Subdirectory	Contents
All installations.	*Tools*		Sentinel LM tools to generate license codes and customize your application with developer-defined data.
		Distctrl	Distribution control program that allows you to create a Windows license code generator for your product distributors.
		Lic	WlscGen stores license types you define in this directory.

Installing the Software on UNIX

Note: A production software license is included and all required libraries and binaries will be encoded with unique customer information during installation. When prompted, enter the serial number included in the Sentinel LM product packaging. This customer information is registered with Rainbow Customer Support and must be specified when support for additional platforms is purchased.

Specific installation instructions depend on the UNIX platform you are using. Please see the printed notice that came with your software for details on installing Sentinel LM for your platform.

The directory *SentinelLM*, will be created in the current directory, and will contain six subdirectories. The files in the directories under *SentinelLM* are:

Sentinel LM Subdirectories

Directory	Files
bin	Contains program files and utilities.
doc	A duplicate of the *README* file. A postscript file of Sentinel LM manuals may be included in some distributions.
examples	Examples that illustrate the various features of Sentinel LM.

Sentinel LM Subdirectories

Directory	Files
include	The include files, which provide the prototypes for various Sentinel LM library functions.
lib	The client libraries.
man	The online Acrobat-format manuals for Sentinel LM.

The *examples* directory contains source code and executables of various licensed and unlicensed executables. Before running these applications, license codes must be generated using the code generator.

Note: The license code generator programs can only be run on Windows. If you mount the CD on a Windows computer, if CD autorun is enabled the license code generator installer will automatically run to install WlscGen.

Tutorials

Sentinel LM includes several tutorials to help you get started using Sentinel LM.

Sentinel LM Start Here Guide for Windows

If you are new to Sentinel LM and would like a tour of some of its features, please see the *Sentinel LM Start Here Guide for Windows* in the Sentinel LM CD case. *Sentinel LM Start Here Guide for Windows* will give you simple hands-on experience using Sentinel LM to protect and license an application.

The UNIX Tutorial

For an example of modifying application code with the Sentinel LM Application Library on UNIX, please refer to the *Sentinel LM Programmer's Reference Manual*.

Installing the Adobe Acrobat Reader

Sentinel LM CD includes Adobe Acrobat online versions of the Sentinel LM manuals. To see the Acrobat files, you need to have Acrobat Reader version 4.0 or higher installed on your computer.

For the Windows versions of Sentinel LM, you may install the Reader from the Sentinel LM release by navigating to the *\Acrobat* directory on the Sentinel LM CD and starting the *AdbeRdr60_enu_full.exe* file. Running that file will install the Acrobat Reader on your hard disk.

For UNIX computers, you can obtain the Acrobat Reader from the Adobe web page, *http://www.adobe.com*. Follow the instructions on that Web page to choose and install the correct Reader for your particular UNIX platform. We also include the Windows version of the Reader on the UNIX release.

Uninstalling Sentinel LM

Windows

To uninstall Sentinel LM on Windows:

1. Click **Start** and point to *Settings* > *Control Panel* > *Add/Remove Programs*.
2. Select **Sentinel LM 7.x.x** and click the **Add/Remove** button.
3. When the uninstallation is finished, click **OK**.

Note: Certain files that were customized by Sentinel LM during installation will not be removed. And, files that you modified or created will not be removed. You may manually remove these files and directories when the uninstallation is complete.

UNIX

On UNIX platforms, Sentinel LM is self-contained in the installation directory. To uninstall Sentinel LM, remove the installation directory.

Chapter 3

Planning Your Application Licensing

Note: This chapter is **Step #3, "Plan Your License,"** of protecting and packaging your application. For a list of these steps, see Chapter 1, "Steps to Protect and Ship Your Application," on page 23.

This chapter contains information on:

- Planning features
- Choosing a license model

Plan Your Use of Features

Sentinel LM provides a great deal of flexibility in how you can license your applications and one way it does this is through the use of features.

At its simplest level, a feature is a way of identifying an application for licensing purposes. One product, one feature!

Let's say your product is an application that helps customers maintain a healthy life style, called **HealthNow**. Licenses that authorize customers to use **HealthNow** identify the application by its feature name (and optionally, its version number). To make the feature name unique to your company and product, you might want to use an abbreviation of your company name, Health Systems, Inc. (HS) and the product name (HealthNow), to produce the feature name *HS_HlthNow*. (A feature name may not contain spaces.)

Whenever your customers run a license usage report on their site, they see your protected application, *HS_HlthNow*.

In this example, a license code for the feature name *HS_HlthNow* authorizes use of the entire application.

Note: If it is important to you that the license codes you distribute to your customers be as short as possible, then you will want to use the predefined license types built into the Windows license code generator, WlscGen, which are optimized to produce short, all-numeric license codes. If you are using these predefined license types, your feature names are limited to two numeric digits.

Multiple Features Per Application

You can also use multiple features to license your application in a more sophisticated way. A single application can have multiple capabilities that you want to license separately. In this case, you can assign multiple features to an application.

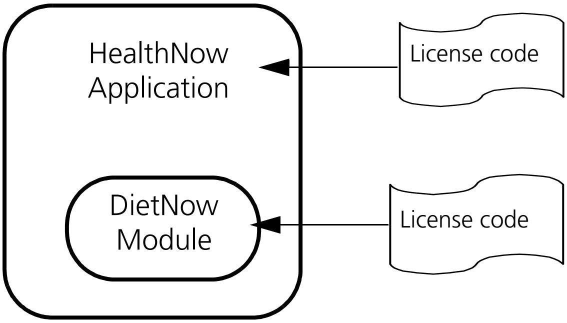
For example, imagine that the new version of **HealthNow** contains two different capabilities within the one application:

- The **DietNow** module, that tracks food intake and reports on calories, fat grams, and sodium consumed.
- The **MenuNow** module, that suggests healthy menus based on your health goals, and logs meals eaten.

By careful use of features, you can license each component separately or in any combination. For example, one customer may license the entire **HealthNow** application, while another may only want the **DietNow** capability.

By assigning the application more than one feature, you can generate licenses that enable different combinations. For example:

- One customer purchases a license that activates the entire **HealthNow** application. The feature licensed is *HC_HlthNow.*
- Another customer purchases a license that activates only the **DietNow** portion of the **HealthNow** application. The feature licensed is *HC_DietNow.*



The license code for feature *HC_HlthNow* authorizes use of the entire **HealthNow** application.

The license code for feature *HC_DietNow* authorizes use of only the **DietNow** portion of the **HealthNow** application.

Planning Use of Features in an Application

For your own application, decide how you want to assign features. Do you want to license the entire application with a single license, or do you want the flexibility of licensing separate components of your application?

Are there key parts of your application that must be available before any other capabilities can be used? Are there parts that can be used and purchased separately? Do you want certain parts of the application to be shared between users, while others are reserved for the use of specific users? Do you want certain parts to be available for a limited time, perhaps as a demonstration, while others are permanently available?

It is even possible to protect the same part of your application with multiple features. You could assign one feature for a time-limited, demonstration license, and the other for a permanent license.

You define feature names while you are protecting your application (either with the Sentinel LM Application Library or with Sentinel LM-Shell).

License Queuing

Consider what will happen when your customer tries to acquire a license but they all are in use. If you want to take care of this case in a graceful way, without requiring your customer to exit the application and try again, you can configure an application to enable license queuing.

If license queuing is enabled, when an application user cannot acquire a license token because none is available, the user is placed into a queue to wait for the next available license. You can select license queuing when you use Sentinel LM-Shell to protect an application, or you can use the Sentinel LM Application Library to include license queuing API calls in your application source code. If you choose to use API calls, you will need to write your own code to handle whether license queuing will be automatic or optional, how long to hold the user in the queue before exiting, what message, if any, to display, and so on.

Note: License queuing is only available for applications that use network licenses.

For information on license queuing, refer to the *Sentinel LM Programmer's Reference Manual* and Chapter 4, "Protecting Your Application with Sentinel LM-Shell," on page 51.

Choose the License Model

Sentinel LM offers a large number of different license types. As you choose the license model you want to use, you will need to choose whether you want to ship your applications to be ready to run on stand-alone computers, networked computers, or both. The way you distribute and package your application will differ slightly based on this decision. For details, see Chapter 7, "Packaging and Shipping Your Application," on page 67.

License models can be chosen at three different points in the process of protecting and shipping your product:

- *Development options* are chosen while the application is being protected (either with Sentinel LM-Shell or the Sentinel LM

Application Library), and can only be implemented during development.

- *License code options* can be configured for each license agreement and/or customer at the time license codes are generated, such as the maximum number of users and sharing of license codes. Upgraded license codes can be generated that activate different features of the protected application without modifying the application.
- *End user options* are configured by your customer and include setting up user alerts, user groups and reservation groups.

These options interact. One licensing model may require special library calls during development and also may require your end user to set up user groups, for example. The point is that some licensing options must be implemented during development, while implementation of others can be deferred to when you generate license codes.

Another thing to keep in mind is that you can combine license models and locking methods to form many new models. For example, you might define a license model that is "shared," and "time-limited." Here is a summary of the basic license models:

License Model Summary (S = Stand-alone computer, N = Networked computer)

License	Description	S/N	Stage
Additive/ Exclusive	An additive license code appends changes to an existing additive license. An exclusive license code overrides any additive licenses for the same application feature/ version.	N	License code
Commuter	Allows end users to "check out" a license authorization from a network and then check it back in when they are done.	N	License code
Capacity	Defines the number of operations that can be executed per hard limit also allows end user to use an extra flag along with hard limit to control numer of applications that can run simultaneously.	N	Development
Feature-based	Licenses by application feature.	S/N	Development

License Model Summary (S = Stand-alone computer, N = Networked computer)

License	Description	S/N	Stage
Held	Specifies a grace period that previous user can request the license again before it is made available to other users.	N	License code
Multi-feature	A single license code that licenses a range of features. Typically used to license an entire set of applications with a single license code.	S/N	License code
Locked	A licensed application is locked to a specific computer. Sentinel LM offers client-locked, server-locked, and client & server-locked.	S/N	License code
Redundant	License is serviced by a pool of redundant license servers.	N	License code
Reserved/ Excluded	Users, groups of users, or computers are given authorization to use the application or excluded from using it.	N	End user
Shared	Allows a user to make use of multiple functions of the same application while consuming only a single license.	N	License code
Site	Authorizes use of application on specific subnet.	N	License code
Time-limited Demo	License expires on a specific date after the license is installed.	S/N	License code
Trial	A time-limited license that uses a trial period of 1 to 120 days that is relative to the date the application is first run, not set to specific calendar dates.	S/N	License code
Version-controlled	Licenses by application version number.	S/N	Development and License code
Log file encrypted	Transaction in the license servers usage log file for a specific license can be encrypted to allow pay per use billing	N	License code and end user
Upgrade	Upgrades the capacity and version of the base license	N	License code and end user

Development Licensing Options

The following licensing options are configured and integrated into the application using client library function calls:

- Feature-based Licensing
- Version-controlled Licensing

Feature-based Licensing

You may sell complex pieces of software composed of several autonomous software components and desire the flexibility of having different numbers of licenses for each component or feature. The feature name, which consists of a short character string, is the smallest unit Sentinel LM can license. Each module of an application can be individually licensed using different feature names.

The license code generator creates an independent license code for each feature selected, enabling autonomous licensing of each feature. Each feature can specify different options such as node locking, number of concurrent copies, license expiration dates, and so on. For example, you might have different applications that contain unique feature names that will be used for permanent license codes specific to each application, while the same applications may also all contain feature name 99 that will be used for demo license codes allowing temporary use of all applications.

To summarize: A single application can use multiple feature names and multiple applications can use the same feature name.

Although the normal case is to have one license code per feature, you can also create a single multi-feature license code that licenses multiple features. One example of using a multi-feature license code is when you want to send a single license code to a customer that is purchasing a suite of applications. For more information on multi-feature license codes, see Chapter 8, “Creating and Upgrading License Codes,” on page 79.

Version-controlled Licensing

Using long license codes, any licensed feature of an application may have a version number, enabling you to implement comprehensive licensing schemes based on release version numbers of your software. The name and version number of a feature are treated together as a unique identifier of the feature. The feature “DOTS,” version “2.0” is therefore distinct from the feature “DOTS.”

License Code Licensing Options

You can configure the following options for the customer based on the license agreement:

- Various methods of locking
- Site licenses
- Shared licenses
- Held licenses
- Additive and exclusive licenses
- Time-limited licenses
- Redundant licenses
- Capacity licenses
- Upgrade licenses

Locking Methods

Sentinel LM provides a variety of licensing schemes on both the application and license server sides.

Note: Following traditional networking terminology, we say that the computer on which the application is running is the *client*, while the computer on which the license server is running is the *server.* However, remember that you can use client locking for a stand-alone computer, and that the license server can run on the same computer as the application.

The method of locking is encrypted in the license code, which enables you to use the same Sentinel LM Application Library calls regardless of locking technique. End users can have their locking method changed or updated with a new license code and will not require a new copy of the licensed software.

Note: If an application or license server is locked to a Computer Id key, the Sentinel System Driver must be installed on that computer.

Unlock Option

Unlocked network applications will run on any computer, provided they receive authorization from some license server. Furthermore, any license server may authorize requests from any application so long as doing so does not violate other terms of the license agreement.

Note: Stand-alone applications also may be unlocked.

Because unlocked applications are completely unrestricted, we recommend you use them with demonstration (time-limited) licenses so that the license will expire. Such time-restricted licenses are useful when shipping beta and evaluation software or as a temporary license for full-featured applications.

Because generating a license code for an unlocked license does not require you to know anything about the computer on which the application will run, such a license code can be shipped with the protected software.

License Server Locking

Applications with server-locked licenses are also permitted to run on any computer. However, only a license server running on a particular computer may grant authorization for the requesting application to run. License servers can be locked to one or more items such as IP address or Computer ID key.

Client Locking

Client-locked applications are authorized to run only on particular computers. A license server may grant authorization for the requesting application to run, but only if the requesting application is running on the appropriate computer and only if doing so does not violate other terms of the license agreement. Licensed applications can be locked to one or more identifying items such as IP address, Computer ID key, or host name. You can specify additional limits on how many copies of your software can run on each client computer. Client locking may also be used for stand-alone computers to lock your application to a specific, non-networked computer.

Client/Server Locking

This is the most restrictive method of locking provided by Sentinel LM. Like client locking, a licensed application can run on a restricted number of computers only. Furthermore, only a license server running on a particular computer may grant authorization to the requesting application. Licensed applications and the license server can be locked to one of several items such as host name, IP address, or Computer ID key. You can specify additional limits on how many copies of your software can run on each client computer.

Site Licenses

Site licensing enables you to specify a set of user-site subnets, which together form a user site. The Sentinel LM license server running at that site will refuse to service calls originating from outside. This option is supported only with unlocked network applications.

Shared Licenses

Shared licenses allow multiple invocations of a licensed application to share the same license or licenses. If license sharing is enabled, the license server determines whether an authorization has already been granted to a requesting application and, if so, it issues an authorization to use the license to the new requestor. You can specify additional limits on how many copies of your software can share a single license. Further license requests will be honored only if more license tokens are available on the license server.

Held Licenses

Held licenses may be useful for applications that are typically short-lived. Once an application is granted a license, the license remains granted for a period of time known as the "hold time." Therefore meaning that the server will hold the license for a held time defined in the license after the license is released by the application. During this hold time, the same user may use the license as often as required, but no other user will be able to use that license. Each time the license is reused, its hold time is extended and the license is held for a time that begins at the latest time of reuse. The grace period is the specified time that begins when the license is released. During the grace period only the original requestor may use the license.

Additive and Exclusive Licenses

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. All other license codes, additive or exclusive, are ignored for that feature/version.

There can also be multiple additive license codes per product in the license file. Unless there are exclusive license codes for the same product, 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 66 and version 5.0.
>
> There may be two additive license codes for 66 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 feature 66 and version 5.0.

Additive license codes are used to increase the number of available license tokens or number of client computers and also to increase the Poloed license capacity.

The *only* items that can differ in all additive license codes for the same feature/version are: number of users, vendor information, and license expiration date.

Note: If you issue a new additive license code to a customer, it must *not* be identical to an existing license on the customer's computer or the license server will ignore it. This is a security measure to prevent a customer from installing the same license code multiple times. Make sure a new additive license differs from existing licenses at the customer site by varying the vendor information, or the number of licenses per client.

Time-Limited Demo Licenses

These licenses are good for a specified time. They are commonly unlocked to provide a demo or temporary license that will run on any computer.

Using the unlocked option enables you to issue licenses ahead of time without having customer information. Because the license is not locked to a particular computer, you can ship the license with your application. Before it expires, your customer contacts you for a permanent license.

Extra Security with Trial Licenses

Sentinel LM uses persistence technology to ensure that trial licenses cannot be re-installed by your customer to be used more than once. Even if the protected application is uninstalled, the computer rebooted, and the application re-installed, the customer will not be able to install the same or a new trial license. Because of this security, you cannot install the same application with a trial license more than once. If you need to update your customer with a new license, use a permanent license.

Note: When your customer purchases a permanent network license after having installed a temporary network trial license for the same feature/version, the permanent license will replace the trial license even if the trial license has not yet expired.

Redundant Licenses

Redundant licenses are license codes that have been configured to be serviced by a pool of redundant license servers. This provides both license server backup and license balancing. In the first case, if a license server goes down, the licenses it was servicing are now serviced by another license server in the pool without interrupting service to the application users. In the second case, license tokens are distributed among the license servers in the redundant license pool, reducing the load on any one license server. For example, if one license code allows 30 concurrent users, 10 license tokens may be allocated to License Server #1, 5 to License Server #2, and 15 to License Server #3.

You will define a license as working with redundant license servers when you generate the license code to send to a customer. For information on creating redundant licenses and license balancing, see Chapter 9, “Using Redundant License Servers and License Balancing,” on page 137 and Chapter 8, “Creating and Upgrading License Codes,” on page 79.

Information for your customers on how to set up and configure a redundant license server pool is contained in the *Sentinel LM System Administrator’s Online Guide.*

Commuter Licenses

Commuter licensing allows a traveler to check out authorization to use an application on a laptop computer when planning to be away from the office network for a while. When returning to the office, the traveler checks the authorization back into the network. If the commuter cannot connect to the office network, a remote commuter license can be generated and sent to the customer.

You define a license as a commuter license when you generate the license code. For information on creating a commuter license, see Chapter 8, “Creating and Upgrading License Codes,” on page 79.

For general information on commuter licensing, see Chapter 10, “Using Commuter Licensing,” on page 155. Information for your customers on how to check a commuter license in and out is contained in the *Sentinel LM System Administrator’s Online Guide.*

Capacity Licenses

Capacity licensing allows the developer to control the number of operations per hard limit. Sentinel LM controls the number of operations by defining the capacity of the operations.

For detailed information on capacity licensing, see Chapter 11, “Using Capacity Licensing,” on page 165.

Upgrade License

Using Sentinel LM ulscgen utility you can upgrade the version and capacity of your base license. Sentinel LM allows you to upgrade version for non-capacity, pooled and non-pooled capacity licenses. Capacity upgrade is allowed only for non-pooled capacity licenses .

For detailed information on generating upgrade license, see Chapter 8, “Creating and Upgrading License Codes,” on page 79.

End User Options

The following sets of options can be prepared by end users:

- Alerts
- Group allocation

Alerts

End users 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 the *Sentinel LM System Administrator's Online Guide*.

Group Allocation Options

The right to run applications can be restricted to particular users, computers, or groups. For more detailed information, see the *Sentinel LM System Administrator's Online Guide*.

Group Reservations

Group reservations give end users 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 are for general use.

On Windows computers, the reservation file editor, WlsGrMgr, can be used to easily create a new or edit an existing reservation file. For information on WlsGrMgr, see the *Sentinel LM System Administrator's Online Guide*.

Including/Excluding Users and 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.

Including/Excluding Teams

Application tokens/capacity can be reserved for certain team(s). The system administrator can ensure that certain teams are always supported on the server.

However, the number of tokens reserved for a group cannot exceed the number of teams specified in the group. If the number of tokens reserved is greater than the number of teams, then the server reserves tokens equal to the number of teams and the rest of the tokens are put in the unreserved pool.

Capacity Reservation for Non-sharing Licenses

In case of non-shared licenses each client consumes one token and the teams are ignored. Therefore even if same machine or user requests a license multiple times, the request is allowed if all other constraints are met.

Each application can request only one token at a time. Since sharing is turned off for these licenses, each individual client will get its own token that is not shared with any other client.

Chapter 4

Protecting Your Application with Sentinel LM-Shell

Note: Along with Chapter 5, "Protecting Your Application with the Application Library," on page 59, this chapter is **Step #4, "Protect Your Application,"** of protecting and packaging your application. For a list of these steps, see "Steps to Protect and Ship Your Application" on page 23.

Sentinel LM-Shell for Windows provides a quick way to protect your 32-bit Windows applications. Sentinel LM-Shell "wraps" your application with a protective shell without modifying program code and is designed to be easy to use, quickly satisfying the most common developer needs. If you want more control or want to use more sophisticated licensing models, you should consider using the Sentinel LM Application Library discussed in Chapter 5, "Protecting Your Application with the Application Library," on page 59.

Sentinel LM-Shell runs on Windows 95/98/ME and Windows NT 4.0/2000/XP/2003 and protects Windows 95/98/ME and Windows NT 4.0/2000/XP/2003 executable files.

Some of the reasons you might consider using Sentinel LM-Shell are:

- You don't have access to the source code for the application you want to protect.

- You want to quickly set up a demo product and don’t need special licensing.
- You want to get familiar with Sentinel LM quickly and easily.

Sentinel LM-Shell has been tested with various programming languages and should be able to protect most applications. The best way to find out if it works with your application is to try it on your application executable files and see if it succeeds. If Sentinel LM-Shell cannot protect your application, refer to Chapter 5, “Protecting Your Application with the Application Library,” on page 59 for information. The Sentinel LM Application Library allows you to embed protection in your application source files.

To begin using Sentinel LM-Shell, we suggest that you first protect a copy of an application you are already quite familiar with (like **Notepad**) to run on a stand-alone computer so you can see how the protected application works with the rest of Sentinel LM. This will also give you a chance to try out the license code generation portion of Sentinel LM as discussed in Chapter 8, “Creating and Upgrading License Codes,” on page 79.

While trying to protect *notepad.exe* if you generate a license, Sentinel LM-Shell will decrement your license meter count.

As you use Sentinel LM-Shell, you will see that it has been designed to make it very easy to license an entire group of programs by assigning all of the programs the same feature name, and also to license each program individually by assigning each program a different feature name. One application for this is, you could provide one temporary, demo license for all of the products on your CD, yet have your customers purchase a permanent license for each product they would like to purchase. This allows your customers to try, before buying your product.

Before Using Sentinel LM-Shell

Decide whether you would like your protected application to run on stand-alone computers, networked computers, or both.

An application protected for both stand-alone and networked computers will attempt to find a stand-alone license first, then if it cannot find one, will request the license from a license server in a network. You can defer this decision until license generation by choosing to protect for both. If your product may someday be sold in network environments, we recommend you protect for both and defer this decision.

Note: If you are upgrading from a previous version of Sentinel LM-Shell, it automatically converts your project file to the new file format when you save it. Therefore, in the new version you will be able to open up previous versions of Sentinel LM-Shell projects and save them, but once you have saved the project you will not be able to open it in the previous version. The new version of Sentinel LM-Shell has new features that cannot be supported in earlier versions.

Using Sentinel LM-Shell

In Windows, start Sentinel LM-Shell by clicking **Start** and pointing to *Programs* > *Rainbow Technologies* > *Sentinel LM 7.x.x* > *English* > *Sentinel LM-Shell*.

Note: Sentinel LM-Shell's user interface is not supported in Windows NT 3.51. Use the command-line version, as described in "Using the Sentinel LM-Shell Command-Line Interface" on page 54.

The Welcome screen asks if you want to **Create a new project** or **Open an existing project**. To set up your protected product, select **Create a new project**. Sentinel LM-Shell window has six tabs - four primary tabs: Files, Assignments, Activation, Protect and two advanced features tabs: User Messages, and Customization.

The **User Messages** and **Customization** tabs are hidden unless you select **Advanced Options** from the **View** menu or click the **Advanced Options** button from the toolbar.

The tabs in the Sentinel LM-Shell window represent the steps necessary to protect your product.

Tip: We recommend that you follow the tabs in the main window in order from left to right when creating a new project. After that, you may access the tabs in any order you wish.

You will find Sentinel LM-Shell very easy to use. The online help that accompanies Sentinel LM-Shell tells you exactly what to do on each tab. You can choose to have the online help appear automatically when you open Sentinel LM-Shell. To enable automatic help, from the **Help** menu, select **Show Automatic Help**.

For details on Sentinel LM-Shell please refer to Sentinel LM-Shell Help. For a quick tour of Sentinel LM-Shell, see the *Start Here Guide* in the Sentinel LM CD case.

Using the Sentinel LM-Shell Command-Line Interface

Sentinel LM-Shell uses a Windows graphical user interface. However, a command-line version is also available. You might want to use the command-line version if you have an automated system for compiling and creating your applications. You can locate the LM32[n][s] executable files at *\Rainbow Technologies\SentinelLM\ 7.x.x \English\LM-Shell.*

You can enter the Sentinel LM-Shell command line directly into your application makefile so that the application automatically gets protected when it is built.

Format:

```
LMS32[N|S] /FLname [options] [path\]progname
[[path\][newname]]
```

where:

Item	Description
LMS32S	Protects applications for stand-alone environments.
LMS32N	Protects applications for network environments.
LMS32	Protects applications for both stand-alone and network environments.
FLname	Licensed feature name (where *name* is up to 24 characters[a]).
Path	Drive and/or subdirectory of the file.
Progname	Program to be protected.
Newname	Resulting protected file, default newname = program name.

[a] If you plan to use short, numeric license codes, you MUST limit the feature name to two numeric characters.

The options available are:

Option	Description
/H or **/?**	Display user help.
/FDname	Demo feature name (where *name* is up to 24 printable characters). If you plan to use short, numeric license codes, you MUST limit the feature name to two numeric characters.
/VDname	Demo version name (where *name* is up to 11 printable characters). Short, numeric licenses do not specify version.
/VLname	Licensed version name (where *name* is up to 11 characters).
/**O**	Overwrites *newname* if it already exists.
/Mname	Defines the message file (default = *slmshell.msg*).
/En	Remind the user when *n* days or less remain before the license expires.
/In	License renewal interval. (default = every 60 seconds)
/Ln [**/C**name]	Specifies license activation mode (*n* = 0, no auto activation; *n* = 1, auto activation). If *n* = 1, /**C**name (*name* = activation configuration file) is required. Option ignored if LMS32N is used.

Option	Description
/**Q**	Enable license queuing option (default = off). Used for network licensing only (LMS32 and LMS32N).
/**R**	Offers opportunity to obtain a trial license. (Valid only if license activation mode=1)
/**D**n	Specifies the number of days prior to license expiration. (Valid only if license activation mode=1)
/**P**n	Server UDP port (*n* = 1-65535; default = 5093).
/**T**n	Number of license tokens consumed or tokens required when requesting a license (default = 1).
/**DLL**name	Name of the DLL for CustomHost ID function.
/**FN**name	Function name for CustomHost ID
@fname	Reads parameters from a file designated by *fname*. If you are using several command line options, you may find it easier to put some or all of the commands in a text file. You then reference the command file's name using the @ option on the command line. You can use multiple command files by specifying the @ option multiple times, or you can include additional @options within the command file itself. For example, the following command invokes Sentinel LM-SHELL using the command line options stored in the files called *PROT1.DAT.* *LMS32[S] @PROT1.DAT FILE.EXE C:\*

To specify a hexadecimal number in any of the numeric options above, prefix the number with a $.

Note: The message file (default slmshell.msg) contains the text of messages displayed to the user of the shelled application, usually when an error occurs. You may edit this file to change the text of these messages. The format of message entries is specified in the default version of this file (slmshell.msg), which is installed in the same directory as the LMS32[N|S] executables.

Example:

```
LMS32S /FLmyapp /VL5.0 /FDmy-demo /O /L1 /Cmyproject.rac
d:\MyPrograms\myapp.exe
```

```
    d:\MyPrograms\Protected\my_protapp.exe
```

This command protects an application called *myapp.exe* located in directory *d:\MyPrograms* and stores the protected copy in directory *d:\MyPrograms\Protected*. You can enable demonstration copies of *my_protapp.exe* by generating license codes with expiration dates for feature “my-demo” You can then issue a permanent license code for feature “myapp” when the user purchases your software. Having a valid license code for either feature will enable the end-user to run the application. This application will use the Client Activator for auto activation, using *myproject.rac* as the activation configuration file.

For more details on client activator, see “Client Activator” on page 63.

Note: When you protect your applications with /L and /C options for Client Activation, you will need to run the Activation Wizard program to generate the activation configuration file, *myproject.rac*.

Chapter 5

Protecting Your Application with the Application Library

Note: Along with the previous chapter, this chapter is **Step #4, "Protect Your Application,"** of protecting and packaging your application. For a list of these steps, see Chapter 1, "Steps to Protect and Ship Your Application," on page 23.

This chapter provides instructions and detailed information on:

- Client library functions
- Compiling your application

Using the Sentinel LM Application Library to embed protection calls in your application source code provides the maximum amount of control and allows you the most flexibility in using licensing models. A detailed description of the Sentinel LM Application Library has been provided in the *Sentinel LM Programmer's Reference Manual*.

Choosing a Language Interface

Adding licensing capability to your application is easy and requires only a little effort. If you are using development languages other than Microsoft C/C++, please check the \Interface subdirectory in the Sentinel LM installation directory, which contains language interfaces for Borland C, PowerBuilder, Microsoft Visual Basic, and Delphi.

Note: The *\MsvcDev\Samples* directory also contains example programs for Microsoft C.

Rainbow continues to develop new programming language interfaces, so if you don’t see the one you need, check with your Rainbow representative for availability information.

To access complete information about Sentinel LM application library and on the how to use the application library to protect your code, we would suggest a detailed study of the *Sentinel LM Programmer’s Reference Manual.*

Chapter 6

License Activation

Note: This chapter is **Step #5 "Plan License Activation."** For a list of these steps, see Chapter 1, "Steps to Protect and Ship Your Application," on page 23.

This chapter contains information on:

- The types of license activation methods you may choose for your protected application.
- The Client Activator, a Windows tool to automate the activation process at the customer site.

License Activation Methods

Sentinel LM offers a wide choice of methods by which your customer can activate a stand-alone product license. Table "License Activation Methods," on page 62 gives information on the different license activation methods available to help you make your choice.

License Activation Methods

Method	Description	Effort	Cost	Security	Comment
Computer ID Key[a]	License code is locked to the Computer ID key so it can be activated before the product is delivered.	Easy	$$$$	High	Highest security using additional queries to the Computer ID key. However, the license code still has to be installed at the customer site
Internet	License code is generated through a web page on the Internet.	Medium	$	High	Set up Sentinel Express Activation Server on your Web page. Ship Client Activator with your product. See "Client Activator" on page 63.
Mail & Fax	Product has a temporary or demo license for use until customer requests and receives permanent license.	Medium	$$	Medium	Temporary license shipped with product. May use Client Activator to provide Fax page with locking code and accept license code when received.
Phone Calls	Operator gives license code over the phone.	Volume dependent	$$$$	High	May use Client Activator to provide locking code and accept license code. User enters license code by keyboard.
Product ID	License code is shipped with product as Product ID. Can be used for temporary or permanent license.	Easy	$	Medium	May use Client Activator to accept Product ID by keyboard and activate application.

[a] Method does not provide end user registration.

Client Activator

Sentinel LM Client Activator for Windows provides an automatic method to exchange the appropriate information with your fulfillment site and activate your Sentinel LM-protected program. You build your product-specific Client Activator and ship it with your protected application to activate either a demo license or a permanent license. Client Activator is intended for use when your customer needs to install a license on a stand-alone or a networked computer.

Sentinel LM Client Activator runs on Windows only.

If you are using Client Activator with a Sentinel LM-Shell protected application, see Sentinel LM-Shell Help.

Note: For information on installing the Client Activator on your development computer, refer to the installation instructions that accompany that product.

Client Activator Process

With the automated Client Activator, the following process activates a Sentinel LM-protected application:

1. After product installation, the first time the applications run, the Activator prompts the end user for company and user information.

2. If needed, the Client Activator generates a locking code and displays it to your customer. (Depending on the activation method you have chosen, the Client Activator may also automatically transmit the locking code to your fulfillment center.)

3. The Client Activator guides your customer through the rest of the activation process:

   ❑ If your activation method is a license code shipped with the application, the Client Activator asks for the code to be entered by keyboard or file.

- If your activation method is by telephone, the Activator prompts for the appropriate information, provides the phone number to contact, and waits for the license code to be entered by keyboard.
- If your activation method is by fax or mail, the Activator prompts for the appropriate information, prepares a fax page which can be printed and sent to the fulfillment site. When the license code is available the user runs the Client Activator program and enters the code.
- If your activation method is through the Internet, the Activator contacts Sentinel Express on your Web site and provides the locking code. After registration and payment verification are received, the license code is returned.

4. The Activator activates the protected application.

For more details on the Client Activation Wizard refer to the Client Activator Wizard help.

Client Activator Operation

The Client Activator includes an Activation Wizard that builds and configures a Client Activator for a specific protected application. The Wizard allows you to choose how the product activation will be presented to your customer, and defines the methods your customer can use to activate the product (faxing a fulfillment center, getting onto the Internet, typing a license code at the Client Activator's prompt, and so on).

Note: When using the Internet for transferring the license code, the Client Activator requires Rainbow's Sentinel Express Activation Server installed on your Web site. See *Sentinel Express Developer's Guide* for complete information.

To build your product-specific Activator, you simply run the Wizard and define your product and activation method. The Wizard builds the Activator, which you ship with your protected application. When the customer installs the application, the Client Activator prompts the customer for the necessary user information and completes the activation.

Client Activator Wizard Process

The Client Activator Wizard initializes and configures the Client Activator for your customers by asking you for the following information:

- Your product name, version, copyright date, and product ID (for example SKU);
- Publisher information, including phone numbers and web site URLs;
- The method or methods of activation you support, keyboard/file, telephone, fax/mail, or Internet;
- Packaging information, such as images to display on the user's screen.

The Wizard builds your product-specific Client Activator, generating the following files:

- *Activator.rac* - Rainbow Activator configuration file. (If you are using Sentinel LM-Shell to protect your products, this file will be renamed by the Shell software to the name of your Shell project file. The protected application will be configured to use that configuration file.)
- *AInst.exe* - Client Activator setup tool.
- *Project.rap* - Rainbow Activator project file. This contains the settings for the Client Activator Wizard for your product.

You must ship the *Activator.rac* and *AInst.exe* files with your product. From the installation of your product, simply run *AInst.exe* to install the Client Activator on the target system. This program is a self-extracting compressed file that installs the Client Activator and required components in the <*Windir*>*Rainbow Technologies**Client Activator**7.x**English* directory where <*Windir*> is *Windows* for Windows 95/98/ME and *Winnt* for Windows NT/2000/XP/2003. At the end of your installation program, run *Activator.rac* to provide activation of your protected product.

For More Information

See the online *Client Activator Wizard help* or *Client Activator Manual* for complete details on building your product-specific Client Activator.

To use the Internet for license activation, full understanding of Sentinel Express License Activation Server is required. See *Sentinel Express Developer's Guide* for complete information.

Sentinel LM-Shell provides a selected set of Activation scenarios for set up during the Shell process. See Chapter 4, “Protecting Your Application with Sentinel LM-Shell,” on page 51 for complete information.

Chapter 7

Packaging and Shipping Your Application

Note: This chapter is **Step #6, "Package and Ship Your Application,"** of protecting and packaging your application. For a list of these steps, see Chapter 1, "Steps to Protect and Ship Your Application," on page 23.

This chapter contains information on:

- How to package your protected application?
- How to modify your application installer to include Sentinel LM files?
- Planning end-user computer fingerprinting.
- What your customer needs to know.

By now, you have protected your application and are ready to package and ship it. If not, see the instructions on protecting your application in Chapter 4, "Protecting Your Application with Sentinel LM-Shell," on page 51 and Chapter 5, "Protecting Your Application with the Application Library," on page 59.

Modifying Your Application Installation

You will need to modify the installer for your application to include the end-user Sentinel LM files. The files you ship differ slightly depending on whether your application is configured for stand-alone or network operation, or both.

In the Sentinel LM directory structure installed on your development computer, the *\Delivery* subdirectory contains a suggested directory structure for the files you ship to your customers. Within that structure are all the files you will ship to your customer, except for your own protected application (*your_app* holds your application files). From this master set of files, you will select the files to ship.

Note: Many directories contain platform-specific subdirectories from which you will select files for the platform on which your application will run (e.g., \Win9x).

For a complete discussion of the Sentinel LM directory structure and what files are contained in it, see Chapter 2, “Installation,” on page 25.

Shipping a Stand-Alone Application

For a stand-alone application, the minimum you need to ship is:

- One or more licensed applications integrated with Sentinel LM client library or protected with Sentinel LM-Shell.
- If you use dynamic linking for your Windows application, you need to ship the client DLL *lsapiw32.dll*. If you choose static linking, you don’t need the DLL. It is optional to send the nonet or the integrated library. Though the name of nonet should be changed to *lsapiw32.dll.*
- If you use the Client Activator utility for auto activation, the following files are required: *ainst.exe* (the Client Activator installer) and *activator.rac* (if you are using Sentinel LM-Shell, this file is renamed to the same name as your Sentinel LM-Shell project file). For details on the Client Activator and its files, see “Client Activator” on page 63.

- If you do not use the Client Activator but use computer fingerprinting, include the fingerprinting utility, *echoid,* with the pre-configured *echoid.dat* file. (For details on echoid and its data file, see "Planning Customer Computer Fingerprinting" on page 73.)
- The license code installation utility, *lslic,* if you do not use the Client Activator for installation and activation.
- Sentinel System Driver if you choose to use the Computer ID key on your end user's computer as the computer fingerprint (Windows computers only).
- An optional *lservrc* file containing a temporary license code or some other file or information sheet containing license codes for licensed features.

Shipping a Network Application

For a network application, the minimum you need to ship is the same as for a stand-alone application (see above) plus the following:

- License server for Windows 95/98/ME, Windows NT/2000/XP/2003, or UNIX. For Windows 95/98/ME or Windows NT/2000/XP/2003 end-user computers, the license server installation program, *setup.exe* (in the *\Delivery\Server\Setup* directory), installs the license server on the customer's computer. This installation program will detect whether your customer is running on Windows 95/98/ME or Windows NT/2000/XP/2003 and install the correct license server for that platform. If your customer is on a Windows NT/2000/XP/2003 computer, the installation program also installs *loadls.exe*, the Windows NT/2000/XP/2003 License Server Loader, under *\Rainbow Technologies\SentinelLM\7.X.X\ English\Delivery\Server\Winnt* directory. The installation program is for your convenience; you may use it or supply your own.

Note: Network licensed applications will require end users to start the license server on the computer specified by the license codes.

- You will also need to include information for your end-user on how to start the license server once it is installed. We recommend you ship the *Sentinel LM SystemAdministrator's Online Guide* to your customer. You may also want to read that guide to become familiar with how your customer sets up the license server.

If you plan to allow end-users to "check out" commuter licensing authorization to laptops or stand-alone computer, you will need to also ship *Wcommute.exe* or *lcommute,* and possibly *commute.dat*. For remote checkout the *rcommute* or *Wrcommute* utilites should also be supplied.

- If you used Sentinel LM-Shell to protect the application and enabled license queuing, you must include *lmsstat.dll*.
- If your end user is using redundant license servers, you will want to ship *wrlftool.exe, rlftool, wlmadmin.exe,* and *lspool*.
- You will need to ship *echouid.exe* to enable your end-user to generate the upgrade code(required for generating the upgrade license) for already existing license .
- You will need to call *vregtool.exe* in your installer, before you ship it to your end-user. It creates registry entries and necessary files required to install your software protected with Sentinel LM, to work.

Note: Applications protected with the network only option of Sentinel LM-Shell do not use the Client Activator for auto activation.

TCP/IP and NetWare IPX/SPX Protocol Support

Your development computer does not have to be configured with the communications protocol that will be used on your customer's computers (TCP/IP and/or NetWare IPX/SPX). Your protected application will work on a computer using either or both protocols.

Tip: If you want to defer the decision to runtime, then either TCP/IP or Netware IPX/SPX should be present.

Here are some guidelines to make sure the application communicates correctly with the network license server. In all cases, make sure that the correct license server is being communicated with, by setting the license server using the LSHOST environment variable or by using the broadcast method of identifying the license server. (See *Sentinel LM System Administrator's Online Guide* for details on license server identification.)

- If only TCP/IP is installed on the customer's computer, the protected application will use TCP/IP to talk to the license server.
- If only NetWare IPX/SPX is installed on the customer's computer, the protected application will use IPX/SPX to talk to the license server. You need to set LSProtocol environment varaible for this
- If both TCP/IP and IPX/SPX are installed on the customer's computer, Sentinel LM uses TCP/IP to communicate with the license server.

Note: If you are using IPX/SPX, the host name of the license server must be the IPX address of the server in this format:
network-node, IPX-address
Where network-node is four hexadecimal bytes and IPX-address is six hexadecimal bytes. For example:
`00-00-12-34,00-00-86-1A-23-A3`

If you need to determine the network address of the license server, use ipxecho on the computer on which the license server resides. (Simply run ipxecho on a computer and it will display the address.) The ipxecho utility can be found at Sentinel LM *\tools* directory. Use the LSPROTOCOL environment variable to control which protocol the client uses to communicate with the license server.

Sentinel LM Tools and Utilities

Sentinel LM offers two categories of tools and utilities: those that are reserved for use by the developer in *\Tools*, and those you can ship to your customer in *\Delivery\admin.net* and *\Delivery\admin.sta*. For a complete list of these, see "Utilities" on page 193.

Note: Of the utilities intended for developer use, there are two that *must not* be shipped to customers: lscgen and WlscGen, the license code generator (Windows command-line and graphic-interface versions, respectively). In fact, distribute them very carefully within your own organization to protect the security of your application.
Other utilities provided include Wechoid, which helps you generate fingerprinting criteria, and lsver, which displays the Sentinel LM version number.

Customer Installation of the Sentinel System Driver

If you are locking your application or the license server to a Computer ID key, you will need to provide the Sentinel System Driver to your customer for installation at your customer's site.

The Sentinel System Driver is required for Windows computers. The drivers are in platform-specific subdirectories in the *\Delivery\Driver* directories. See the file *\Delivery\Driver\readme.pdf* for instructions for installing the driver.

Customer Installation of the Sentinel LM License Server

An independent installation program has been provided for Windows 95/98/ME and Windows NT/2000/XP/2003 end-user computers that installs the license server on the end-user computer. This installation program is called *setup.exe*, and is located in the *\Delivery\Server\Setup* directory. You may use this installer or provide your own.

If a copy of the license server has already been installed at your customer's site, perhaps for an earlier version of software or for a different application, then you only need to install a license server if you have a newer version of it. The way you determine the version number of the license server depends on the license server platform. For Windows 95/98/ME, run the license server and select the **About** command on the **Help** menu. For UNIX computers, run the license server; it will display its version number.

Information on configuring the license server on a customer's computer is contained in the *Sentinel LM System Administrator's Online Guide*.

Planning Customer Computer Fingerprinting

Unless you choose an unlocked license model, you may want to lock your application or the license server to a particular computer. To do so, you will need to determine the locking code of the computer based on its fingerprint. (The fingerprint distinguishes one computer from another.) If you are using auto activation, the Client Activator takes care of this for you, by displaying the locking code during license activation.

If you are not using the Client Activator, your customer will use the echoid utility to determine the computer locking code. The echoid utility generates a locking code from the fingerprint. Your customer then gives the locking code to you and you use it to generate a license code for that specific computer.

Before you ship your protected application, you need to plan how this fingerprinting will be done.

Choosing Fingerprint Criteria

You choose the criteria of the computer fingerprint. It is important that you consider this decision carefully, because the different criteria differ in security benefits and risk of failure. (The more secure the fingerprinting, the less likely the customer is to defeat the licensing; the higher the risk of failure, the more likely it is that the customer will not be able to obtain a legitimate license.)

For example, locking to the Host Name gives less security, because your customer can change the Host Name on another computer to a duplicate name, thus gaining the ability to run the application on multiple computers. However, the Host Name also has a very low risk of failure, since it is not likely to change.

On the other hand, locking to the disk ID gives greater security, but the hard disk has a greater risk of failure than the Host Name (that is, it is more likely that the hard disk will fail and the customer will have to replace it, leaving your customer unable to obtain a license).

In picking what criteria will make up the fingerprint, you need to balance your need for security against possible inconvenience to your customer.

How to Select Fingerprint Criteria

echoid uses the *echoid.dat* file that you ship to your customer which defines what criteria will make up the fingerprint. To specify what criteria to use for the computer fingerprint, use a text editor like Windows **Notepad** to modify the copy of the file *echoid.dat* located in the *\Delivery\Admin.net* or *\Delivery\Admin.sta* directory to specify a hexadecimal or binary value that defines the criteria you want to use. The fingerprint criteria are bit values that are ORed to form a byte. For example, the hexadecimal value 0x4 selects the hard disk ID. The value 0x2 selects the IP address. To select both the hard disk ID and the IP address, add the values together and enter the sum at the end of the file: 0x6. You will ship this modified file to your customers.

Note: The Client Activator sets up a hex value for the fingerprint criteria in the activation configuration file. If you determine that you want to define your own value, you can edit the configuration file.

For the most recent list of the specific fingerprint criteria supported, refer to the *echoid.dat* file contained in the *\Tools* directory, which contains comments listing all criteria and their values.

Here are the fingerprint criteria supported at this time.

Supported Fingerprint Criteria

Criteria	Definition	Availability	Set in.	Hex Value
ID PROM	Processor serial number or ID.	Typically RISC processors in UNIX.	Hardware	0x001
IP address	Network address of computer.	TCP/IP	Software	0x002

Supported Fingerprint Criteria

Criteria	Definition	Availability	Set in.	Hex Value
Disk ID	Hard drive serial number.	Windows computers only.	Software	0x004
Host name	Name of the host.	UNIX and Windows NT/2000.	Software	0x008
Ethernet address	Unique Ethernet card ID.	Network using Ethernet. Windows and UNIX computers.	Hardware	0x010
NetWare IPX	IPX network address of computer.	IPX/SPX	Software	0x020
NetWare Serial Number	Each NetWare server has a unique serial number.	IPX/SPX	Software	0x040
Computer ID key	Hardware protection device.	Windows computers.	Hardware	0x080
Custom	Defined by your application.	N/A	N/A	0x100
Processor Number	Intel Pentium III processor number.	Windows	Hardware	0x200

Selecting a Primary and Secondary Fingerprint

To add to fingerprinting flexibility, you can specify a primary and a secondary fingerprint using two different groups of fingerprinting criteria. You do this by placing two numeric values in the *echoid.dat* file. For network licenses, the license code generator accepts two locking codes when you lock a non-redundant license server, allowing you to generate a license code which verifies first one fingerprint and then, if the first one fails, verifies a second. Or, you can simply choose which fingerprint to use at license code generation time.

Why would you specify two fingerprints in the license code?

Suppose you lock to Hard Disk ID and the end-user has to replace the hardware. You can avoid generating another license by providing a second fingerprint.

If you place two numeric values in the *echoid.dat* file, when the customer uses echoid, the customer will see two locking codes displayed. When you use the license code generator, you know that the first value given to you by the customer is based on the primary fingerprint, and the second value is based on the secondary fingerprint.

You must be careful when choosing fingerprinting criteria to make sure you are accomplishing your goal. We recommend that you include elements in both fingerprints that are unlikely to fail (for example, Host Name and IP address), and also include in each fingerprint a different, higher-security element such as Ethernet card address and Disk ID.

The table below gives some suggested settings for primary and secondary fingerprints to provide various licensing options:

Suggested Primary and Secondary Fingerprints

Suggestion	Windows		UNIX	
Security Option	**Group1**	**Group2**	**Group1**	**Group2**
	Computer ID key	Disk ID	Computer ID key	ID PROM
Stand-alone and Network Option	Disk ID	IP address and Ethernet address	ID PROM	ID PROM
Laptop Option	Disk ID	IP address and Ethernet address	ID PROM	ID PROM
Backup Option	Disk ID	Host Name	ID PROM	Host Name

For information on how the license code generator accepts multiple locking codes based on two separate fingerprints, see Chapter 8, “Creating and Upgrading License Codes,” on page 79.

Tip: When choosing a fingerprint for a laptop computer, we recommend selecting the Disk ID.

What Your Customer Needs to Know

You may wish to send your customer the *Sentinel LM System Administrator's Online Guide*, which contains information on utilities and options that can be used by end users. In addition, you will want to give your customer written information on the following topics:

- How the customer can contact you to obtain new and upgrade license codes.
- If you are not using auto activation with the Client Activator, how the customer installs the first license for the newly-installed protected application. Are you shipping temporary license codes that do not lock to a particular computer, or will the customer need to use echoid to compute the fingerprint of the computer and then send you the locking code?
- How the customer receives license codes from you.
- How the customer installs the license.
- How the customer purchases upgrade license codes.
- Where the customer can get help if problems occur.

How Does Your Application Runs on the Customer Site?

For an overview of how your customer interacts with your protected application, refer to Chapter 1, "What Does Licensing Mean to Your Customer?," on page 21.

Chapter 8

Creating and Upgrading License Codes

Note: This chapter is **Step #7, "Create License Codes,"** of protecting and packaging your application. For a list of these steps, see Chapter 1, "Steps to Protect and Ship Your Application," on page 23.

The license code generator is the heart of the Sentinel LM product. It helps you to generate the license codes that your customer will install to activate your protected application. This chapter contains information on:

- The Windows graphical user interface version of the license code generator, WlscGen. See "Using the License Code Generator – WlscGen" on page 90.
- The Windows (DOS) command-line version of the license code generator, lscgen. See "The License Code Generator Command-line Version – lscgen" on page 112.
- The Windows graphical user interface of the upgrade license code generator.
- The Windows (DOS) command-line version of the upgrade license code generator, ulscgen.

- Special license code generation choices such as creating redundant licenses, commuter licenses, capacity licenses and log file encryption. See “Special License Code Options” on page 97.
- The Distribution Control Program that allows you to create a special version of the Windows or DOS license code generator for use by your distributor.

Sentinel LM provides two forms of the license code generator: a Windows-graphical user interface version and a DOS command-line version. Both run on Windows 95/98/ME or Windows NT/2000/XP/2003 only. And both require a hardware license meter key that authorizes their use.

Sentinel LM developers will find the Windows-interface version of the license code generator especially useful because of its two operation modes: clerk and administrator mode. In administrator mode, you can set up templates for your most common licensing situations. These templates allow you to pre-set many of the licensing choices and parameters for the clerk who will be generating the license codes. In clerk mode, a clerk uses these templates to generate license codes, filling in only the information that is customer or order-specific.

Before You Start

This section contains some information about license code generation you should be aware of before you start.

License Code Generator Startup

To generate license codes, you must have a license meter key installed on the computer on which you are going to run the license code generator.

You will only be able to run the license code generator if you received a Sentinel LM license meter key from Rainbow Technologies. To install the license meter key, connect it to the parallel or USB port of your computer (depending on which type of hardware key you received)as discussed in the next section.

⚠ The license codes you create will decrement the license meter key. An unlimited, unlocked license code will decrement the license meter key by 250, so be careful not to empty the license meter key; if you do, you will need to purchase a new one.

The license code generator in the evaluation version of Sentinel LM is customized with a demo developer ID. That means any license codes you create with it will only work for applications protected with the evaluation version of Sentinel LM.

The License Meter Key

The Sentinel LM license meter key, which meters the Sentinel LM license code generator, must be attached to the parallel or USB port of your IBM-compatible PC, depending upon the type of key. The parallel port license meter is the red key included in your Sentinel LM product.

If you have a parallel port key, attach the end of the Sentinel LM license meter labeled COMPUTER to a parallel printer port. Tighten the screws to connect the key securely to the port. If a printer is connected to the computer, attach a shielded printer cable to the outside connector of the license meter key.

The Sentinel System Driver, used by the license meter key, is automatically installed on your computer when you install Sentinel LM.

Note: If you manually install the system driver and license code generator on another computer and are using a USB license meter key, be sure to install the driver with the setup \USB option.

How Does The License Meter Key Decrement?

The license meter key meters the number of license codes generated. The initial license meter key shipped with Sentinel LM contains two license counters: one counter keeps track of *trial licenses* (relative time-limited licenses that use a trial period of 1 to 120 days); another keeps track of all other types of licenses.

Fixed Decrement Values

Some types of license codes decrement the license meter key by a fixed amount:

- Stand-alone or network license codes that are not locked to a particular computer and are not time-limited decrement the license meter key by 250.
- Network license codes that are for an unlimited number of users and are not time-limited decrement the license meter key by 250.
- Stand-alone license codes that are locked to a specific computer decrement the license meter key by 1.

Note: Network licenses that are not time limited that also enable an unlimited number of users or do not lock to any computer, decrement the license meter by 250. If you are not aware of this, it is very easy to accidentally exhaust your license meter key.

Calculated Decrement Values

Other types of license codes cause the license meter key to be decremented using the following equation:

Decrement Value = (Days active / 365) * Number of users

Here is more information on the elements of this formula for different types of licenses:

- Network or stand-alone license, unlocked, time-limited: Number of users set to 250, *Days active* set to number of days between start date and end date (or trial period if it is a trial license).
- Network or stand-alone license, locked, infinite number of users, time-limited: *Number of users* set to 250, *Days active* set to number of days between start date and end date (or trial period if it is a trial license).
- Network or stand-alone license, locked, finite number of users, not time-limited: *Days active* set to 365.

- Multi-feature license codes: the license meter key is decremented separately for each feature using the formula above. See the examples below.

Examples of Decrementing the License Meter Key

For example, if you create a license for 10 people to use the application for five weeks, the license meter decrements:

35 / 365 * 10 = 0.9

or 1, since the license meter decrements by a minimum of one for each license created.

As another example, if you create a 60-day trial license that does not lock to a specific computer, the license meter decrements:

60 / 365 * 250 = 41.09

or 42, since the meter key rounds up to the next whole number.

Here is an example of a multi-feature license code:

A stand-alone, 30-day trial license code is unlocked for an infinite number of users and authorizes use of three features. For each feature, the license meter key decrements:

30 / 365 * 250 = 20.5

or 21. Adding together all three features = 21 + 21+ 21, or 63.

When the License Meter Key Is Empty

When the trial license counter is exhausted, WlscGen looks for an optional subscription license code file. See below.

Note: Sentinel LM uses up the license meter key trial license counter before it begins to use the optional software license file. See below. If there is no subscription file, the main counter is decremented.

Also, see “Cascading License Meter Keys” on page 85 for information on attaching more than one license meter key.

Subscription License File

The Sentinel LM subscription license file allows you to generate an unlimited number of licenses. See the next section for details

Characteristics of the Subscription License File

The characteristics of the subscription license file are determined at the time the file is purchased by the developer and include:

- The length of time the subscription license file can be used by the developer (usually one year)
- The feature/version that licenses can be generated for
- Whether the subscription license file can be used for trial licenses or normal licenses or both
- Whether the subscription license file can be used for stand-alone licenses, network licenses or both
- If the subscription license file supports network and normal licenses, then whether it can be used for redundant, capacity and/or commuter licenses

Note: Subscription licensing is not allowed for upgrade licenses.

Contact your Rainbow sales representative for information on purchasing a subscription license file.

Using the Subscription License File

The subscription license file supports generation of normal licenses or trial licenses or both. To make use of the subscription license file, place it in the same directory as WlscGen or lscgen (or use the lscgen -s option to specify the location of the subscription license file). The default file name is lservrc.

Tip: The Sentinel LM license meter key must be attached to the computer generating the license codes, even though it will not be charged. Subscription license files generated for Sentinel LM 7.2.0 and later will not support normal licenses generated using pre-Sentinel LM 7.2 WlscGen or lscgen.

Supplementing the Trial License Counter: Optional License Code File

You may optionally purchase a license code file that supplements the license meter key trial license counter. This license code file is available for separate purchase on an annual subscription basis and allows you to generate an unlimited number of trial licenses during a one year period for a specific feature/version and developer ID. Contact your Rainbow sales representative for information.

At start up, the license code generator looks for the license meter key. When found, it then searches for the optional license code file:

1. It looks at the environment variable LSERVRC for the location of the license code file. Meaning that if the environment variable is set to LSERVRC then preference will be given to the environment variable even if you have lservrc file in the same directory in which you have *lscgen.exe*.

2. If the variable LSERVRC is not found, the license code generator looks at the first license code of the file *lservrc* in the same directory as the license code generator.

For information on how the license meter key meters license codes generated, see “How Does The License Meter Key Decrement?” on page 81.

Cascading License Meter Keys

You can use more than one license meter key by attaching multiple keys together. (This is called *cascading* license meter keys.) This is how Sentinel LM handles multiple license meter keys attached to the parallel and USB ports:

- The number of license tokens remaining in the license meter key reported by WlscGen or lscgen is the sum of all license tokens left in *all* license meter keys attached. (If the license meter keys have trial license counters, a separate value is reported that is the sum of the trial license counters in all the attached license meter keys. Another value is reported that is the sum of all main license counters in all the attached license meter keys.)

Note: lscgen supports only 2 license meter keys.

- When you generate a license code with WlscGen or lscgen, Sentinel LM begins decrementing the license meter key that is physically nearest to the computer parallel port. When that license meter key is empty, Sentinel LM begins decrementing the next closest license meter key, and so on. For USB keys, Sentinel LM decrements the keys in the order of their assigned port numbers.

About License Codes

When you install Sentinel LM in non-evaluation mode, an exclusive developer ID (or vendor code) is built into the code generating utility as well as the client library. This means that license codes you generate are labeled as yours, and cannot be used to license other developers' applications. This also protects your applications from being licensed by other developers' license codes.

When using Sentinel LM in evaluation mode, you must never distribute license codes made with an evaluation version of WlscGen or lscgen or distribute applications protected by an evaluation version of Sentinel LM to your customers, because any other developer using Sentinel LM in evaluation mode will be able to issue (and sell) licenses for your applications.

License codes are uppercase alphanumeric or all-numeric strings. For added security, you can add custom encryption/decryption routines for the network licenses to the code generator and the license server.

Any license codes generated can be used on any Sentinel LM-supported platform. For example, if you create a license code on a Windows 95/98/ME computer, that license code can be used on a UNIX computer or a Windows NT/2000/XP/2003 computer.

A license code can be:

- Long: Long codes contain as much information as needed.
- Short: Short codes contain less information and cannot support certain licensing options, but have the advantage of being easier to generate and easier to communicate to customers.
- Short, numeric: Short, numeric license codes are particularly acceptable to customers because they are all-numeric and about credit-card number size. Because short, numeric license codes are optimized for a small size, they do not support special license code features such as commuter and redundant licenses. To create a short, numeric license code, use the optimized license types predefined in WlscGen and lscgen.

You can configure the license code licensing options for the customer based on the license agreement. For example:

- Application-server locking
- Shared licenses
- Trial licenses
- Commuter licenses
- Capacity licenses
- Additive and exclusive licenses
- Time-limited licenses
- Redundant licenses
- Log file encryption level

Please see Chapter 3, “Planning Your Application Licensing,” on page 35 for more information.

You can also specify that a license code be readable or encrypted. (Readable license strings contain plain text that gives information about the license agreement, and can be read by your customers so they can verify license provisions.) For details on readable license codes, see Appendix A, “Readable License Codes,” on page 227.

Predefined License Types for Short, Numeric License Codes

Before you begin to create your own license codes, take a look at the predefined license types available within the license code generators.

Several of the license models most frequently used by developers are built into both WlscGen and lscgen, and are optimized to produce all-numeric, short license codes. Not only are the numeric, short license codes easier to transmit to your customers (and easier for them to type), but using the predefined license types presets most of the license information for you, making creating a license code quick and easy.

In WlscGen create a **New** license template and further choose to create a **Full license template**. On the **License Features** window select **License code type** as **Predefined - Multi Feature** or **Predefined - Single Feature.** The predefined license types are listed in the **Predefined license code types** drop-down list.

In **lscgen**, you see the predefined license types if you select “[2] Short, Numeric Code” when asked to specify license code length.

Here are the predefined license types:

- **1 to 30-Day Standalone Demo -** Produces a temporary license code for a stand-alone computer that will expire in a specified time limit of 1 to 30 days after the application protected with the license is first run on the customer’s computer. (This is a relative time-limited (or trial) license, which calculates the expiration date based on the date the protected application is first run.) You may optionally specify an end date to limit the life of the license regardless of when it is installed. The license is exclusive and the application is not locked to a specific computer.

- **1 to 30-Day Network Demo -** Produces a temporary license code for a computer for which license management is done on the network. The license will expire in a specified time limit of 1 to 30 days after the after an application requests the license from the customer's license server. This is a relative time-limited (trial) license, which calculates the expiration date based on the date the protected application is first run. You may optionally specify an end date to limit the life of the license regardless of when it is installed. The license is exclusive and the application is not locked to a specific computer nor is it locked to the license server.
- **Absolute Standalone Demo -** Produces a temporary license code for a stand-alone computer. An absolute time-limited license contains a set expiration date. The **License Duration** default field is set at **Never** or the license will expire on a specific month and year chosen by you. The license is exclusive and the application is not locked to a specific computer.
- **Absolute Network Demo -** Produces a temporary license code for a computer for which license management is done on the network. An absolute time-limited license contains a set expiration date. The **License Duration** default field is set at **Never** or the license will expire on a specific month and year chosen by you. The license is exclusive and the application is not locked to a specific computer nor is it locked to the license server.
- **Locked Standalone Demo -** Produces a temporary license code for a stand-alone computer. The **License Duration** default field is set at **Never** or the license will expire on a specific month and year chosen by you. The license is exclusive and the application is locked to a specific computer.
- **Buy Standalone Product -** Produces a permanent license code for a stand-alone computer. The application is locked to a specific computer and the license is exclusive.
- **Buy Network Product -** Produces a permanent license code for a computer for which license management is done on the network. The

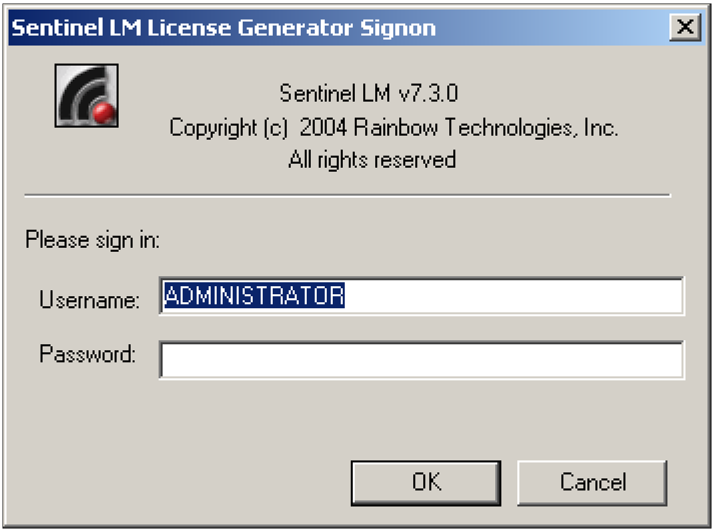
license server is locked to a particular computer and the license is exclusive.

Because these license types have been optimized to produce short, numeric codes (about credit card number size), certain options have been disabled that would increase the license code size. For example, you cannot specify a version number. Short, numeric license codes also cannot be redundant licenses or commuter licenses, capacity licenses or use log file encryption.

Note: The size of the license code increases if you include vendor information or create a multi-feature license code.

Using the License Code Generator – WlscGen

WlscGen is the Windows-interface version of the license code generator. To begin, click **Start** and point to *Programs* > *Rainbow Technologies* > *Sentinel LM 7.x.x* > *English* > *Generate Licenses*. The first thing you see is the logon screen:



Sentinel LM License Generator Signon

Sentinel LM v7.3.0
Copyright (c) 2004 Rainbow Technologies, Inc.
All rights reserved

Please sign in:

Username: ADMINISTRATOR

Password:

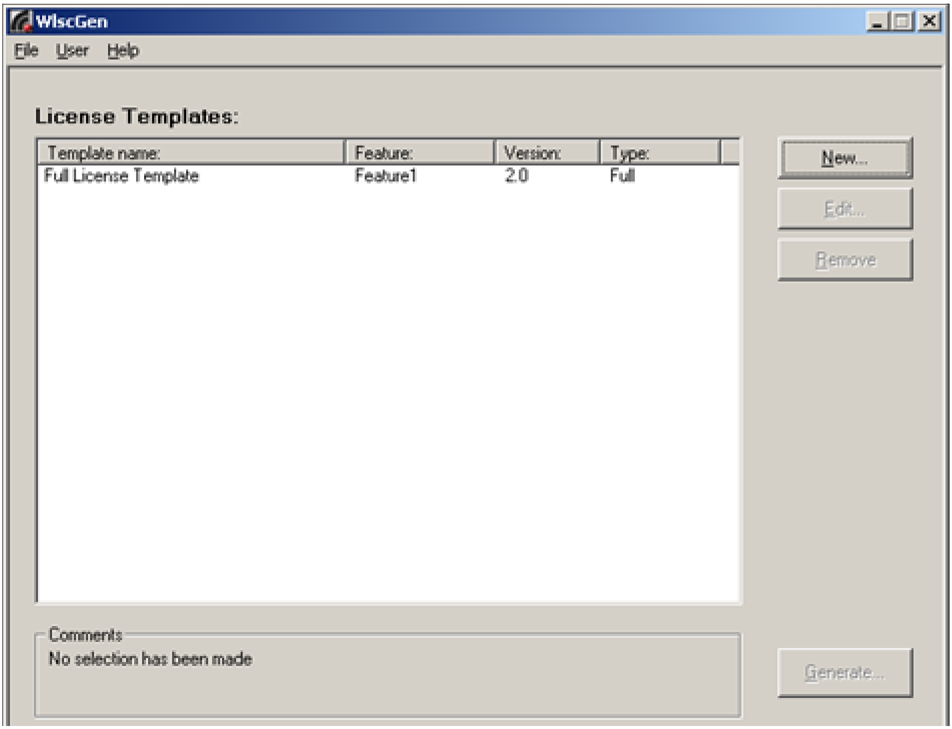
License Generator Log On

The First Time You Use WlscGen

The first time you use the license code generator, log on as user ADMINISTRATOR and leave the password blank.

Note: The blank administrator password takes you to the License Templates screen.

When you are finished, click **OK** to see a list of license templates. You may see something like this:



License Templates:

Template name:	Feature:	Version:	Type:
Full License Template	Feature1	2.0	Full

Comments
No selection has been made

WlscGen License Templates

Note: If you are running the License Code Generator for the first time, no templates would be availble.

One of the first things you will want to do is to set a password for ADMINISTRATOR to prevent unauthorized use of the license code generator. To add a password, select the **Edit** command from the **User** menu. When presented with a list of users, select ADMINISTRATOR, click the **Edit** button and then enter a password.

Administrator and Clerk Modes

The **WlscGen** administrator can:

- Define WlscGen users (add, modify or delete users).
- Create license templates, called *license types* (and modify or delete license templates).
- Generate license codes.

The WlscGen clerk can:

- Input customer information into the **Generate License Code** screen.
- Generate license codes.

Creating and Managing Users

The first thing the license code generator administrator will do is create users. When you create a user, you define who can use the license code generator and whether that user will have administrator or clerk privileges. Depending on your organization, you may want to define multiple clerks with different passwords, or just define a generic “Clerk” user with a single password.

Under the **User** menu there are three options:

- **New**– Select **New** to create a new user. Also specify if the new user is an administrator or clerk and whether the user must enter a password to log on.
- **Edit**– Select **Edit** to change the details of an existing user. You may add or change a password or you can change the user type.

- **Delete**– Select **Delete** to remove an existing user from the database.

Only administrators can create, modify, or delete users.

Creating License Types

Before a license code can be created, you must have one or more license types defined. (Before defining your own license types, see if any of the predefined license types will suit your needs. See “Predefined License Types for Short, Numeric License Codes” on page 88.)

Introduction to License Types

You can think of the license type as a template that defines a particular type of license. You can create three types of license types: Long, short and short, numeric. In many cases, you will find that you set many of the license parameters in the same way every time you create a license code. Creating license types allows you to predefine many of the parameters for different categories of licenses.

Here are some examples of license types:

- You have a new product, VideoEFX, that you are planning to release to potential buyers as a demo. You could create a template for a demo license that uses these parameters (among others):

Sample Parameters: Demo License

Licensing Option	You Select
License code type	Custom
Custom code length	Short
First feature name	02
License type	Stand-alone
Application-server locking	Unlocked
Treatment of multiple licenses	Exclusive
Trial licenses	Set trial and trial period

In the license code type, you can either choose **Custom**, **Predefined-**

Single Feature, or **Predefined-Multi Feature**. Select a **First feature name** that identifies your product.

- You also want to define a license type for VideoEFX for customers who wants to buy your product. You could create a template for a VideoEFX permanent license that sets:

Sample Parameters: Permanent License

Licensing Option	You Select
License code type	Predefined-Single Feature
Predefined license code types	Buy Network Product
Feature name	01

The predefined license code type means you want your purchased product to run in a network environment The license server will be locked to a specific computer on the network.

Of course there are many other parameters you can set in a license type, but the examples above give you an idea of why templates are useful. Once you define the demo license type for a particular product, the WlscGen clerk can use it to generate license codes, and most of the decisions have already been made. To make it even easier for the clerk, while you are creating templates you can predefine some parameters, such as:

- Default number of licenses - You fill in the values for fixed parameters (such as the hard limit on number of licenses), and the clerk cannot change them if **Fixed** is selected.
- Default locking - You fill in the locking criteria for the primary server and the client locking criteria. For example, you may want the server locked to the IP address or the computer ID key.

Creating a New Template

If you decide to create your own templates, here are your choices as displayed on Figure "WlscGen License Templates," on page 91:

- **New–**Click this button to create a new license template. Select the **Full license template** option to generate license code. You may select **Upgrade license template** to create upgrade license template. See "Upgrade License Code Generator Using WlscGen" on page 127 for more details on generating upgrade license code.
- **Edit–**Click this button to edit an existing template file.
- **Remove–**Click this button to delete an existing template file.
- **Generate** –Click this button to generate a license.

Alternatively, right-click on the license template list and select the **New** option from the context menu.

For details on creating a new template or editing an existing one, we suggest you to refer to the Windows License Code Generator Help.

To access Windows License Code Generator Help, click **Help Topics** in the **Help** menu or press F1.

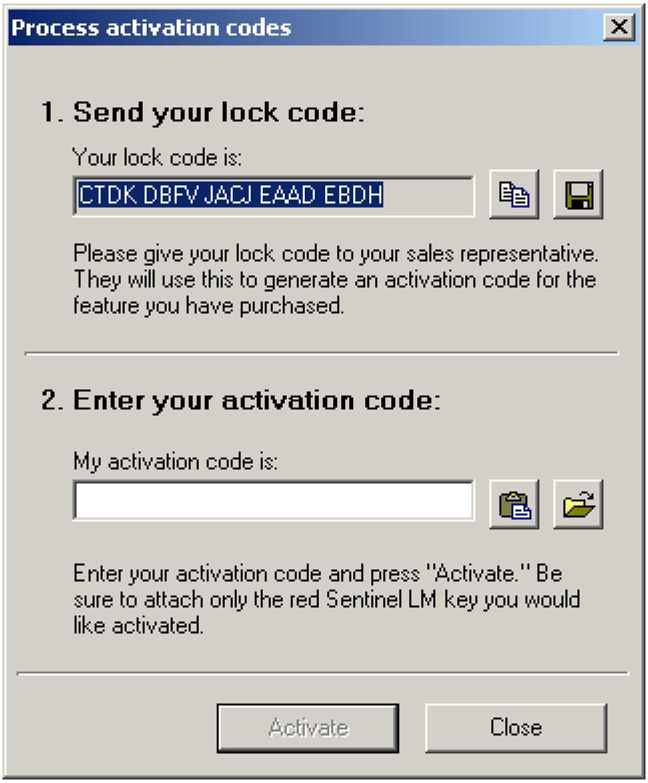
Upgrading Sentinel LM 7.x License Meter Keys

If you have Sentinel LM 7.2.x unlimited license key you can begin to use the Sentinel LM 7.3. You will not be able to use Sentinel LM 7.3 specific features and should contact Rainbow Technologies technical support for assistance. If you have Sentinel LM limited license meter keys for versions prior to 7.3.x, you will need to upgrade them to that of Sentinel LM 7.3.x. Although you can access WlscGen by attaching your Sentinel LM 7.x keys but you will not be able to use it until you upgrade your existing keys using **Process activation codes** from the **Help** menu. When you insert a pre Sentinel LM 7.3.x key, a message box is displayed that informs you of your unsupported license meter key and also directs you to the **Process activation codes** option in the **Help** menu.

As an administrator, WlscGen allows you to upgrade your existing Sentinel LM 7.x license meter keys to 7.3.x compatibility.

To convert your existing Sentinel LM 7.x keys:

- Select **Process activation codes** from the **Help** menu. **The Process activation codes** screen is displayed.



1. Send your lock code:

Your lock code is: CTDK DBFV JACJ EAAD EBDH

Please give your lock code to your sales representative. They will use this to generate an activation code for the feature you have purchased.

2. Enter your activation code:

My activation code is:

Enter your activation code and press "Activate." Be sure to attach only the red Sentinel LM key you would like activated.

Process activation codes

- The **Process activation codes** screen displays the lock code for your key, currently in use. Copy or save the displayed code.

- Next send the lock code to Rainbow Technologies technical support to obtain the activation code for your Sentinel LM 7.x license meter keys.

Tip: You will find a different lock code for the same key each time you invoke the Process activation codes.

- Once you receive the activation code, enter it in the **My activation code is:** text box. You can also paste it from the clipboard or open it from a file.
- Next, click **Activate** to activate your key.

Special License Code Options

Using either WlscGen or lscgen, you can select special license code options that provide additional security and convenience to you and your customers.

Commuter Licenses

Sentinel LM allows you to define a network license code as a *commuter*, or portable, license code. Your customer can temporarily “check out” an authorization to use a commuter license on a laptop computer when planning to be away from the home network, and then check it back in when returning to the office.

Note: A commuter license expires in a maximum of 30 days, it is not necessary to check it back in unless the license server is running low on commuter license tokens. Also note that when a commuter lic expires it automatically gets checked in.

Commuter licensing can only be enabled for network license codes, and is not supported for predefined (short, numeric) license code types.

For more information on commuter licensing, see Chapter 10, “Using Commuter Licensing,” on page 155.

Creating a Commuter License

To specify that a network license code can be used as a commuter license:

- In WlscGen, select the **Allow network licenses to commute** check box on the **Optional License Features** screen.
- In lscgen, use the -O option to select commuter licensing.

  See “Using lscgen” on page 113 for information on the lscgen options.

Redundant Licenses

If your customers have large network installations they would need license server backups, an optimized speed with which a customer requires license tokens and reduce traffic for a particular license server.

To provide license server backup and license balancing, Sentinel LM provides redundant license servers. For a full discussion of redundant license servers and license balancing, see Chapter 9, “Using Redundant License Servers and License Balancing,” on page 137. Redundant licensing can only be enabled for network license codes, and is not supported for predefined (short, numeric) license code types.

Creating a Redundant License

To specify that a network license code can be used as a redundant license (that is, that it will be used with redundant license servers):

- In WlscGen, we strongly recommend that you select the **Enforce majority rule when using redundant servers** check box. (For information on majority rule, see Chapter 9, “Using Redundant License Servers and License Balancing,” on page 137.)

  When generating the license code, select the **Use redundant servers** check box on the **Generate License Code** screen.

- In lscgen, use the -U option to define a redundant license.

  Use the -maj option to select majority rule. See “Using lscgen” on page 113 for information on the lscgen options.

Using a Redundant License

After you define a license code as redundant using lscgen or WlscGen, the rest of the process is up to your customer. Your customer will set up redundant license servers and maintain the redundant license server pool by using the following utilities:

- Wrlftool and WlmAdmin for Windows
- rlftool and lspool for DOS and UNIX.

These tools will also be used by your customer to redefine license token allocation among the redundant license servers, and to add and delete redundant license codes to the redundant license server pool.

For information on setting up redundant license servers and on using the redundant license server pool utilities, see the *Sentinel LM System Administrator's Online Guide*.

Log File Encryption

Log file encryption allows developers to encrypt their log files to secure usage data. Sentinel LM developer may want to charge the customer based on actual license use ("pay per use" billing). Information on each license transaction is stored in the license server usage file at the customer site, and this information can be used to estimate charges if the entries in this log file are secure; that is, unchanged (or unseen) by the customer.

Log file encryption is set as part of the license code (that is, at license code generation time), and defines how or if transaction entries for that license code will be encrypted in the license server usage log file.

Log file encryption can only be enabled for network license codes, and is not supported for predefined (short, numeric) license code types. There are five different levels of encryption available:

Encryption Level

Level	Description
0	• End-user defined. • This level leaves the encryption choice up to your customer. The customer can then specify the encryption level using the -lfe license server startup option. (If you specify end-user defined encryption, and your customer chooses not to select an encryption level at license start up, Sentinel LM uses “no encryption; tag each record” as the default encryption.) Note that the customer only defines the encryption level for license codes set to end-user defined encryption; end-user encryption is ignored for license codes set to other encryption levels by the developer.
1	• No encryption. • Transaction data will be logged for this license code with no encryption.
2	• No encryption; tag each record. • Transaction data will be readable by the end user, but tampering with or deleting an entry will be detected.
3	• Encrypt usage only. • Transaction data will be readable by the customer except for license usage data (which will be encrypted), and tampering with or deleting an entry will be detected.
4	• Encrypt entire record. • All transaction data for the license code will be encrypted, and tampering with or deleting an entry will be detected

The vusage and lsusage utilities react differently when detecting tampering depending on the level of detection. See the *Sentinel LM System Administrator’s Online Guide* and in this manual, see “Display Encrypted/ Readable License Server Log File (vusage)” on page 212.

Defining Log File Encryption Levels

To specify the level of encryption you want to use when entries in the license server usage log file are written for this license code:

- In WlscGen, select the encryption level for this license code in the **Log file encryption level** drop-down box on the **Optional License Features** screen.
- In lscgen, use the -enc option to select the encryption level, where you follow the -enc option with one of the numbers (0-4) discussed in the section above. See “Using lscgen” on page 113 for information on the lscgen options.

Using Log File Encryption

One method a developer might use for implementing pay per use billing is to specify a log file encryption level when generating a license code, then asking the customer to send a copy of the log file to the developer at the end of each month. The developer would then use the vusage to display a report of how many licenses were used by the customer and would charge accordingly.

vusage can read and display encrypted log files. Each license server session is assigned a sequence number which is contained in the log file. The developer should review the session numbers displayed by vusage to verify that the customer did not tamper with a log file or omit sending a log file. For example:

- Log file #1 contains session numbers 10, 11, 13, and 14. Session number 12 is missing, indicating that the log file has been tampered with.
- Log file #2 contains session numbers 15, 16, and 17.
- Log file #3 contains session numbers 19 and 20. Session number 18 is missing, indicating that a log file is missing or that one of the other two log files has been tampered with.

For each billing period, the developer should record the last session number in the last log file and then compare the first session number from the log file of the next billing period to see if a gap has occurred.

For details on using vusage, see “Utilities” on page 193.

For information on the format of the license server usage log file, see the *Sentinel LM Programmer’s Reference Manual*.

Your customer can refer to the *Sentinel LM System Administrator’s Online Guide* for information on using the license server -lfe option to set user-defined encryption.

Note: Log file entries are developer specific, and one developer’s vusage utility cannot read another developer’s encrypted log file entries.

Details on License Code Generator Choices (for WlscGen and lscgen)

Here are details on the items you can select in the license code generator to define your license codes. Included are the license code generator default values and ranges.

When using WlscGen, short, numeric codes are created when you use predefined-single feature and predefined-multi feature license types.

Details on License Code Generator Choices

Short or Long Codes	
Description:	License codes are 16 characters or longer uppercase alphanumeric or all-numeric strings.The code generator will generate long, short or short, numeric license codes. For added security, you can add custom encryption/decryption routines for network licenses to the code generator and the license manager.
Short Code:	Contain less information than the long code and cannot support certain licensing options, but has the advantage of being easier to generate and easier to communicate to end users.
Long Code:	Contain as many characters as needed.

Details on License Code Generator Choices

Short, Numeric Code:	Generates numeric strings only and requires minimal information from the user. Short, numeric codes contain the least information.
Locking Criteria	
Description:	Licenses generated for locked applications require information about the computer to which the license server or application will be locked. A license can be Server Locked, Client Locked, Client and Server Locked or Unlocked. The locking criteria is discussed in the Windows License Code Generator Help.
Short Code:	Yes
Long Code:	Yes
Short, Numeric Code:	Yes
Subnets	
Description:	Site-licensed applications can be locked to IP subnets. A subnet is specified using a wildcard notation. For instance, the subnet notation, 123.4.56.* refers to the subnet comprising all hosts whose IP addresses share the first three sub-addresses with the specified notation. Up to four wildcard can be used to specify a site, one in any sub-address position. Each notation can specify a group of subnets, for example, 123.*.56.*. Supported for long unlocked and server locked license codes.
Short Code:	Not allowed.
Long Code:	Yes. The maximum value for long code is 7 specifications.
Short, Numeric Code:	Not allowed.
Default Value:	*.*.*.*.
Vendor Information String	
Description:	Any piece of information can be encoded into a license code. This information can be retrieved later through a client library function call. This capability is useful for keeping track of distributors or implementing a variety of licensing schemes. It is recommended that you do not use the # symbol, since this symbol is sometimes used as a comment character. The vendor information string is added to the end of the license code. All characters must be in the ASCII range of 32-127.

Details on License Code Generator Choices

Short Code:	Not Allowed
Long Code:	Yes - Maximum value is 395 characters.
Short, Numeric Code:	Yes - Maximum value is 395 characters.
Default Value:	Empty string.
Soft Limits	
Description:	Soft limits are used only for alerts on an application. Whenever the total number of licenses issued exceeds the soft limit, an alert is generated, and action is taken as specified by the alerts configuration script. For more information, see the *Sentinel LM System Administrator's Online Guide*.
Short Code:	Not allowed.
Long Code:	Yes - Maximum units is 32766.
Short, Numeric Code:	Not allowed.
Default Value:	Unlimited
Treatment of Multiple Licenses	
Description:	An additive license code adds users to an existing license code. For example, if the customer has a network license for 10 users and wishes to purchase an additional 5 licenses, you then generate a new additive license for 5 users. An exclusive license code will completely replace the old license with a new recreated license. For example, if your customer has a network license for 10 users and wishes to purchase an additional 5 licenses, you then issue a new network license for 15 users. The customer will then need to install the new 15 user license, which will replace the old 10 user license. If end users are provided with multiple license codes for a single feature, Sentinel LM allows you to choose whether the codes should be combined or only one of the license codes should be accepted. The following rules summarize the license server's behavior when it encounters multiple license codes for the same application. First, all expired license codes are ignored. Then:

Details on License Code Generator Choices

	a) Exclusive license code using redundant license servers and the action taken: • Overwrites any exclusive or additive non-redundant license code for the same feature and version. • Overwrites any redundant additive license code for the same feature and version. • Overwrites any redundant exclusive license code for the same feature and version.
	b) Additive license code using redundant license servers and the action taken: • Overwrites any exclusive or additive non-redundant license code for the same feature and version. • Overwrites any trial license codes for the same feature and version. • Combines any redundant additive license codes for the same application if the additive license codes differ only in the number of users, license expiration date, or vendor information. All other licensing information (such as type of locking, hold time, and so on) must be identical. In this case, the total number of users for all of the additive license codes will be used, the vendor information will be taken from the last valid license code, and the expiration date will be the earliest expiration date for all of the additive licenses.
	c) Exclusive license code not using redundant license servers and the action taken: • Overwrites any non-redundant additive license codes for the same feature and version. • Overwrites any non-redundant exclusive license codes for the same feature and version, if the new license code is being dynamically added or if the new license code is written as the last license code in the license file.
	d) Additive license code not using redundant license servers and the action taken: • Combines with any non-redundant additive license codes for the same feature and version, if the additive license codes differ only in the number of users, license expiration date, or vendor information. All other licensing information (such as type of locking, hold time, and so on) must be identical. In this case, the total number of users for all of the additive license codes will be used, the vendor information will be taken from the last valid license code, and the expiration date will be the earliest expiration date for all of the additive licenses. • Overwrites any trial license codes for the same feature and version.

Details on License Code Generator Choices

	e)Additive license code for pooled capacity license and the action taken: • An additive license will get added only if the additiveness criteria is met otherwise it will overwrite the existing license. • For a pooled capacity license only feature version of two additive licenses should match and capacity and hard limit of both the license will get added.
Short Code:	Both exclusive and additive are allowed.
Long Code:	Yes, both exclusive and additive are allowed.
Short, Numeric Code:	Exclusive only.
Clock Tampering Action	
Description:	You can specify whether the application should run if clock tampering is detected. This check will only be done when the license server starts up, but the license server will not exit on detection of tampering. Only those license strings that specify they want the check will be denied if tampering is detected. Other features will continue to be served by the license server. Even if someone sets the clock back after starting the license server, and then dynamically adds a tamper-sensitive license string, the license server will detect it and throw the license string out. When the license server accepts a license string at startup but detects later that the clock has been set back, it does not grant a license for the feature. Please see “Integrated Cleaning Tool (lsclean and lscgcln)” on page 202.
All codes:	You can specify whether or not a license will be issued if clock tampering is detected.

Details on License Code Generator Choices

Application-Server Locking	
Description:	The desired type of locking must be selected from the following: • Client & Server locking - If an application is licensed to operate in this mode, the application can only run on a specified set of computers on a network. The license server is also locked to a specific computer, although it may be different from the clients. • Client locking - A client-locked application restricts the client application to run on a specific set of computers, but the license server may run on any computer. • Server locking - If an application has server locking, it can run on any computer on the network. Although the license server is restricted to a specific computer. • Unlocked - With unlocked licenses, this last restriction on the license server is also removed, enabling the application and/or the license server to run on any computer.
Short Code:	Yes
Long Code:	Yes
Short, Numeric Code:	Only client-locking, server-locking, and unlocked are supported.
Licenses per Client Node	
Description:	For each client locked and client&server locked node, the number of copies running on each computer is controlled. This is valid only for client & server locked or client locked licenses. This is an extra per-host restriction in addition to the overall number of licenses.
Short Code:	Not allowed.
Long Code:	Yes - Maximum units is 30 or unlimited.
Short, Numeric Code:	Not allowed.
Default Value	Unlimited.

Details on License Code Generator Choices

Sharing Criteria	
Description:	Enables shared licenses and specifies the sharing criteria. The sharing criteria identify which client-related information the license server must consider in order to determine the eligibility of a new client to share a license already issued to an existing client. You have considerable flexibility in defining “sameness.” Licenses can be shared based on user name, X-display ID, or host name of the computer from which the requests originate. For complex situations, it is possible for each application to provide the license server with a vendor-defined shared ID while requesting a new authorization license. The license server will use that ID to determine whether an active license was already issued to any other application with the same shared ID. Each application has complete flexibility in computing its shared ID.
Short Code:	Yes
Long Code:	Yes
Short, Numeric Code:	Not allowed.
Default Value	No sharing.
Sharing Limit	
Description:	Instead of having an unlimited number of users share a single license, this limit forces the issue of a new license when the sharing limit has been reached for a particular license. For unlimited sharing, use No-limit.
Short Code:	Yes - Maximum units is 30 or unlimited.
Long Code:	Yes - 254 or unlimited.
Short, Numeric Code:	Not allowed
Default Value	Unlimited.
Licenses Type	
Description:	Stand-alone applications require a separate license. Network and stand-alone licenses cannot be used interchangeably. You have to know whether the customer is going to run an application on a network or on stand-alone computers. A license server running in network mode will reject stand-alone licenses, and vice versa.

Details on License Code Generator Choices

Short Code:	Yes
Long Code:	Yes
Short, Numeric Code:	Yes
Default Value	Network
Hold-time Selection	
Description:	If held licenses are enabled, the hold time can be specified through license codes, or it can be specified through a client library function call. “Sameness” is defined as the same user name and the same application host name and is independent of process ID. Held licenses may be used in conjunction with license sharing, in which case the “sameness” criterion is the same as that used for sharing.
Short Code:	Yes - Supported only through API only. Use VLSetHoldTime. Maximum 15 hours.
Long Code:	Yes - 15 hours.
Short, Numeric Code:	Yes - Supported only through API. Use VLSetHoldTime. Maximum 15 hours.
Holding Criteria	
Description:	Specifies whether the hold period is in the license code or set in the application.
Short Code:	Yes - Supported through API. Use VLS_Hold_Vendor
Long Code:	Yes - Maximum is 15 hours.
Short, Numeric Code:	Yes - Supported through API only. Use VLS_Hold_Vendor
Lifetime of a License	
Description:	Specified in minutes. The time scale and granularity can also be selected. The range of license lifetimes is from one minute to 15 hours.
Short Code:	Yes - Maximum units is 15 hours or infinite.
Long Code:	Yes - Maximum units is 15 hours or infinite.
Short, Numeric Code:	Yes - Maximum units is 15 hours or infinite.

Details on License Code Generator Choices

DefaultValue	No default.
Number of Licenses	
Description:	An number of applications that will be able to run concurrently with no restriction.
Short Code:	Maximum number of licenses is 254.
Long Code:	Maximum number of licenses is 32766.
Short, Numeric Code:	Maximum number of licenses is 126.
DefaultValue	No limit.
License Start Date	
Description:	If a beginning date is chosen for the license, the application will not run before that date.
Short Code:	Not allowed.
Long Code:	Yes - Maximum year is 2130.
Short, Numeric Code:	Not allowed.
DefaultValue	01/01/2003.
License End Date	
Description:	If an ending date is chosen for a license, the license will become invalid after that date. The license server will reject requests pertaining to the license and the application will not be able to run unless a new license code is provided. If NEVER is selected, the application license does not expire.
Short Code:	Maximum is Never. Specifying the day of the month is not allowed.
Long Code:	Maximum is Never.
Short, Numeric Code:	Maximum is Never. Specifying the day of the month is not allowed.
DefaultValue	Never.

Details on License Code Generator Choices

Feature Name	
Description:	All applications must have a name by which they will be identified, consisting solely of digits and letters. Do not include spaces in feature name. All characters must be in the ASCII range of 32-127.
Short Code:	11 characters.
Long Code:	24 characters.
Short, Numeric Code:	2 numeric digits.
DefaultValue	No default.
Feature Version	
Description:	An application may have an optional version number, consisting of digits and letters. All characters must be in the ASCII range of 32-127.
Short Code:	Not allowed.
Long Code:	11 characters.
Short, Numeric Code:	Not allowed.
DefaultValue	Blank.
Capacity	
Description:	Capacity licensing expands on hard limit and sharing by allowing you to specify the number of operations a token user can perform. Capacity licensing also allows the end user to use an extra flag along with hard limit to control numer of applications that can run simultaneously.
Short Code:	Not allowed.
Long Code:	The following specification style are available for capacity: • 0 - Multiple of 1(s), maximum 1022 • 1 - Multiple of 10(s), maximum 10220 • 2 - Multiple of 100(s), maximum 102200 • 3 - Multiple of 1000(s), maximum 1022000 • 4 - Multiple of 10000(s), maximum 10220000
Short, Numeric Code:	Not allowed.

Details on License Code Generator Choices

Default Value:	Infinite

The License Code Generator Command-line Version – lscgen

lscgen provide three way of accepting inputs from user : command line options, global default file, and interactive input.

When used in command-line mode, the code generator will first check if a required input is available from the entered command line. If available, the corresponding entry in the global defaults file will be ignored. If not available on the command-line, the global defaults file will be checked. You will be prompted for any remaining unresolved required input. If an error occurs with a specified command-line option, the code generator will terminate *without* checking the global defaults file.

The command-line version of the license code generator can be used at the operating system command prompt. See Table “Details on License Code Generator Choices,” on page 102.

Setting Options for the License Code Generator

Use the CODEOPTS environment variable to set the name of the file containing various global options for the Sentinel LM license code generator. (See “Global Defaults File” on page 120 for details on these options.)

Set the variable to a file name:

```
SET CODEOPTS = filename
```

If this environment variable is not set, the license code generator searches for a file named *codeopts* in its own directory.

Use the NEWCODES environment variable to define the name of the license code generator output file:

```
SET NEWCODES = filename
```

If this environment variable is not set, the license code generator creates a file named *newcodes* in its own directory.

The examples above are for Windows 95/98/ME and Windows NT/2000/ XP/2003. Add these statements to your *autoexec.bat* file. For information on how to set environment variables on other platforms, see the *Sentinel LM System Administrator's Online Guide*.

Using lscgen

The command-line version of the license code generator, lscgen, can be invoked with the following options. Using the -H, -h, or ? option alone will generate a usage summary.

The following applies to the command-line options:

- The Tokens-per-node value applies to all locked clients.
- Symbol | indicates a choice of 2 values.
- All client and server lock information should be in hexadecimal.
- There should be no spaces between option letters C, S, R, and their arguments.
- Options that specify file paths must contain a space between the option and the file path.
- The format X/Y/Z... takes the values as 0/1/2... respectively.

Note: When you license more than one feature during a session, the command-line options carry over from one licensing to the next. For example, if at first you chose a server-locked license, then all features licensed in the session will have server-locked licenses.

The following table briefly describes the Code Generator command-line options. For details see Table "Details on License Code Generator Choices," on page 102.

Code Generator Command Line Option

Option	Description
-a	Specifies the day the license expires (1 - 31).
-A	Specifies either network or stand-alone licensing. • 0 = Network licensing • 1 = Stand-alone licensing
-b	Specifies short, short, numeric, or long license code. • 0 = Short • 1 = Long • 2 = Short, numeric If you request a short, numeric code (value 2), you must follow that value with a colon and the number of the license type you want to use (example, -b *2:7*). • 1 = 1-30 day stand-alone (relative demo license • 2 = 1-30 day network (relative demo license • 3 = Absolute stand-alone demo license • 4 = Absolute network demo license. • 5 = Locked stand-alone demo license • 6 = Buy stand-alone product license • 7 = Buy network product license.
-B	Specifies the number of subnets used for site licensing. Long codes only.
-c	Used to specify the type of locking desired: • 0 = Server Locked • 1 = Client&Server Locked • 2 = Unlocked • 3 =Client Locked
-C	• Format is: **-C***n<lock-selector>:<locking-code>*. • Used for client-locked applications to specify the locking code. Do not include spaces. For example, -C*10x2:0x3455*.

Code Generator Command Line Option

Option	Description
-D	Specifies additive or exclusive licenses. • 0 = Additive license (number of keys will get added) • 1 = Exclusive license (will override additive licenses)
-e	Name of the license configuration file.
-enc	Specifies the log files encryption level for network licenses: • 0 = End-user defined. This level leaves the encryption choice up to your customer. The customer can then specify the encryption level using the -lfe license server startup option. (If you specify end-user defined encryption, and your customer chooses not to select an encryption level, Sentinel LM uses “no encryption; tag each record” as the default encryption.) Note that the customer only defines the encryption level for license codes set to end-user defined encryption; end-user encryption is ignored for license codes set to other encryption levels by the developer. • 1 = No encryption. Tag each record. Transaction data will be logged for this license code with no encryption. • 2 = No encryption, tag each record. Transaction data will be readable by the customer, but tampering with or deleting an entry will be prevented. • 3 = Encrypt usage only. Transaction data will be readable by the customer except for license usage data (which will be encrypted), but tampering with or deleting an entry will be prevented. • 4 = Encrypt entire record. All transaction data for the license code will be encrypted, and tampering with or deleting an entry will be prevented. Tampering with or deleting an entry will not be prevented, but can be detected if the appropriate usage utility is run on the log file. Notice log file encryption is not available for stand-alone licenses.
-E	Specifies the feature name.
-F	Specifies the soft limit or no limit. NOLIMIT is a valid value.

Code Generator Command Line Option

Option	Description
-g	Specifies the location of an optional file containing global licensing options.
-G	Specifies the type of shared licenses. • 0 = No sharing of license • 1 = User - name based sharing • 2 = Hostname based sharing • 3 = x - display based sharing • 4 = Application defined sharing
-h	Specifies the type of held licenses. • 0 = Not allowed • 1 = Client application specifies hold time • 2 = The licensing code specifies hold time
-H	Specifies the system attribute used for primary license server locking information. Applicable to non-redundant license servers. Format is: -H<*lock-selector*>*:*<*locking code*>.
-i	Lists any proprietary vendor information. Long codes only.
-I	Used for server-locked applications, specifies the attribute used for locking. Secondary license server locking information. Applicable to non-redundant license servers. Format is: -I<*lock-selector*>*:*<*locking-code*>.
-j	Specifies license holdtime units (1, 10, 30, or 60 minutes) used with -J < *hold minutes*> option. • 0 = Multiple of 1 minute(s), maximum 15 minutes • 1 = Multiple of 10 minute(s), maximum of 150 minutes • 2 = Multiple of 30 minute(s), maximum of 450 minutes • 3 = Multiple of 60 minute(s), maximum of 900 minutes
-J	Specifies the license hold time, in units specified by -j.
-k	Specifies license lifetime units (1, 10, 30, or 60 minutes) used with -K <*lifetime minutes*> option. • 0 = Multiple of 1 minute(s), maximum 15 minutes • 1 = Multiple of 10 minute(s), maximum of 150 minutes • 2 = Multiple of 30 minute(s), maximum of 450 minutes • 3 = Multiple of 60 minute(s), maximum of 900 minutes

Code Generator Command Line Option

Option	Description
-ktype	Format is: -ktype <*0*> or -ktype <*1*>. Notice, 0 will prompt for single feature and 1 prompts for multi feature. This option is normally used with the number of features switch.
-K	Specifies the license lifetime in units specified by -k. NOLIMIT is a valid value.
-L	Specifies whether clients are locked to the same license server in the case of client and server locked licenses (y, n).
-m	Specifies the month the license expires (1 - 12 or Jan. - Dec).
-maj	Majority rule. (Only for redundant license servers.) If majority rule flag is set and majority of license servers are not running, client request will not be served.
-M	Repeats prompting. If set to 'Y' then after generating the first license it prompts if you wish to generate more license but if set to 'N' then it exits after generating the first license.
-n	Used to restrict the number of concurrent users. Specifies the number of licenses. NOLIMIT is a valid value.
-n1:, -n2:	Number of users for multi-features (*-n1:5* = 5 users for feature 1, *-n2:3* =3 users for feature 2 and so on).
-nfeat	Specifies the number of features for a multi feature license.
-nserv	Number of redundant servers. Maximum set of servers is 11.
-N	Used with client locking, this option provides the number of client nodes.
-o	Name of the output file. By default, the code generator appends new license codes to the file, *newcodes*, in the local directory. If the output file does not exist, it will be created. The location of the output file can also be specified by the environment variable, *NEWCODES*.
-O	Commuter License. Used to generate keys for the traveling clients. Commuter licensing allows end users to "check out" an authorization from a network served license pool and "check it in" when they are done. The value of the switch has to be given as either 'Y' for commuter license or 'N' for non-commuter license.

Code Generator Command Line Option

Option	Description
-p	Specifies the number of licenses per node. NOLIMIT is a valid value.
-P	Specifies whether or not an application should run if time tampering is detected. 'Y' or 'N' has to be given as the value of the switch.
-Q	Specifies the sharing limit (1-255). NOLIMIT is a valid value.
-r	Specifies the number of challenge-response secrets. An additive security mechanism for the server (challenge) and client (response). Maximum number of secrets is 7.
-Rn	Text of the secret(s). For example, *-R2text*.
-s	Name of the software license file.
-serv1:, -serv2:	Locking information for redundant license servers.
-S	Specifies subnet information for the subnet(s) used for site licensing. May be repeated once for each subnet. Long codes only. For example, *-S1 204.*.1.*, -S2 205.*.1.**
-T	Specifies the day the license begins (1 - 31).
-U	Redundant Licenses. Allows you to equally disperse the license tokens throughout the network. Redundancy allows the total number of tokens to remain available to the enterprise even if one or more license servers fail. This license balancing allows the developer's end user to set up an initial distribution of tokens among different sites, and the Sentinel LM license servers will automatically adjust the distribution of the tokens to match the actual usage patterns of the tokens across the enterprise.
-v	Specifies the feature version number. Long codes only.
-V	Specifies the month the license begins (1 - 12 or Jan. - Dec).

Code Generator Command Line Option

Option	Description
-ver	Selects the version of Sentinel LM to create a license code. For example, *-ver 5* causes a license code compatible with Sentinel LM version 6.0 to be created. Valid arguments are: • 5 = Sentinel LM versions 6.0.x • 6 = Sentinel LM versions 6.1 and 6.2. • 7 = Sentinel LM version 6.3 - 7.2. If omitted, a Sentinel LM version 7.x-compatible license code will be created.
-w	Specifies the license type: • 0 = Encrypted • 1 = Expanded readable • 2 = Concise readable
-W	Specifies life in days of license relative to date of license installation.
-y	Specifies the year the license expires (1999 - 2126). NEVER is a valid value. Note: If you use the -b <*license code*> option and the -y <*expiration*> option, you must also specify a quarter in which the license will expire. For example: • 1 = March 31 • 2 = June 30 • 3 = September 30 • 4 = December 31
-Y	Specifies the year the license begins (2003 - 2130).
-cap_flag capacity-flag	Defines whether the license is a non-capacity or capacity pooled or capacity non-pooled license. Valid values are 0 or 1or 2. • 0 = VLScg_CAPACITY_NONE • 1 = VLScg_CAPACITY_NON_POOLED • 2 = VLScg_CAPACITY_POOLED

Code Generator Command Line Option

Option	Description
-cap_units	Defines the capacity units. Valid values are 0,1,2,3,4: • If capacity_units is 0, capacity shall be From 1 through 1022, (Multiple of 1). • If capacity_units is 1, capacity shall be From 10 through 10220, (Multiple of 10). • If capacity_units is 2, capacity shall be From 100 through 102200, (Multiple of 100). • If capacity_units is 3, capacity shall be From 1000 through 1022000, (Multiple of 1000). • If capacity_units is 4, capacity shall be From 10000 through 10220000, (Multiple of 10000).
-cap	Specifies capacity. • 0 - Capacity shall be From 1 through 1022, (Multiple of 1). • 1 - Capacity shall be From 10 through 10220, (Multiple of 10). • 2 - Capacity shall be From 100 through 102200, (Multiple of 100). • 3 - Capacity shall be From 1000 through 1022000, (Multiple of 1000). • 4 - Capacity shall be From 10000 through 10220000, (Multiple of 10000).

Global Defaults File

License codes may often require common information, such as feature names, license lifetimes, and sharing criteria. Common information can be stored in a defaults file and referenced by the code generator during license code generation. The code generator first looks for a file specified by the command line. If that is not found, it looks for a file specified by the environment variable, CODEOPTS. Finally, it refers to the file, *codeopts*, in the local directory.

The following is a sample global defaults file for the code generator:

```
1     # Long code
66    # Feature name
2.12  # Prompt for version
```

```
0     # Non-capacity license
N     # Non-trial license
0     # A network license
1     # Log file encryption level
0     # An additive license
      # Vendor data string - None
1     # Username sharing
1     # Username sharing - Lower boundary
1     # Lifetime specification
150   # License lifetime
0      # Hold time - None
7      # Number of challenge/response secrets-Maximum
secret1
secret2
secret3
secret4
secret5
secret6
secret7
1999    # Start date year
1       # Start date month
1       # Start date day
1999    # Expiration date year
1       # Expiration date month
3       # Expiration date day
N       # Issue commuter licenses? - No
0       # Hard limit
0       # Soft limit
0       # Application/server locking (Float)
N       # Redundant servers - No
0x1E    # Primary lock selector - server main
0x56Ac1 # Primary locking code - server main
0x02    # Secondary lock selector - backup server
0x27226 # Secondary locking code - backup server
7       # Subnets - Maximum (7)
192.168.1.*
192.168.2.*
193.*.*.*
193.200.1.*
```

```
195.123.2.*
200.200.*.*
201.213.1.*
Y   # Clock tampering - issue keys if detected? - Yes
Y   # Generate License? - Yes
2   # License string type - Concise readable
N   # Generate more licenses? - No
```

A defaults file should contain one entry per line, each entry corresponding to an interactive input. For those inputs not defined in the defaults file, an exclamation point (!) should be used. Comments can be entered with the pound sign (#) in the first position. White space is ignored. Blank lines will be interpreted as new line inputs. Every line is significant.

Note: You may overwrite settings in the defaults file by specifying an option on the command-line.

Interactive Input

If you use lscgen without a global defaults file or command line option, you will use it in interactive mode.

The following example is an interactive session selecting a long license code using Sentinel LM. This example shows a session without using command-line options or a global defaults file. If there were command-line options and/or a global defaults file used, some or all of the prompts would not appear.

```
Sentinel LM 7.3.0 License Generation System
Copyright (C) 2004 Rainbow Technologies, Inc.


Please type the desired responses at the prompts below.

Desired length and options of licensing code format -
  [0] - Short code
  [1] - Long code
  [2] - Short Numeric Code
  Please enter choice: 1
```

```
Feature name (Any printable characters): myapp
Feature version number (Any printable characters): 1.0
Desired capacity type of licensing code-
[0] - No capacity
[1] - Non-pooled capacity
[2] - Pooled capacity
Please enter choice: 0

Do you want to generate trial license? (Y/N) :N

Please enter the desired client library behavior -
  [0] - Network licensing
  [1] - Stand-alone licensing
  Please enter choice: 0
Enter log file encryption Level (0 - 4)[1]:0

Combining property of licensing code -
[0] - Additive license (number of keys will get added)
[1] - Exclusive license (will override additive licenses)
Please enter choice: 1

Proprietary vendor information (Any printable characters): wrs

Type of shared licenses -
  [0] - No sharing of authorization keys
  [1] - User-name based sharing
  [2] - Hostname based sharing
  [3] - X-display based sharing
  [4] - Application-defined sharing
  Please enter choice: 2

Sharing limit [1 - 254] or <Enter> for no limit: 1

Token lifetime specification -
[0]- From 1 through 15 minutes,(Multiple of 1 minute(s))
[1]- From 10 through 150 minutes,(Multiple of 10 minute(s))
[2]- From 30 through 450 minutes,(Multiple of 30 minute(s))
[3]- From 60 through 900 minutes,(Multiple of 60 minute(s))
Choose the specification style that suits you best:0

Lifetime of each token (in minutes only) or <Enter> for no
limit: 15

Type of held licenses -
```

```
  [0] - Not allowed
  [1] - Client application specifies hold time
  [2] - The licensing code specifies hold time
  Please enter choice: 2

Token hold time specification -
[0] - From 1 through 15 minutes,(Multiple of 0 minute(s))
[1] - From 10 through 150 minutes,(Multiple of 1 minute(s))
[2] - From 30 through 450 minutes,(Multiple of 2 minute(s))
[3] - From 60 through 900 minutes,(Multiple of 3 minute(s))
Choose the specification style that suits you best: 1
Hold time of each token (in minutes only): 20

Number of challenge-response secrets or <Enter> for none: 1

Secret text #1 (Any printable characters): secret1

License start information -
  License start year (2003 - 2130) or <Enter> for a
  current license: 2003
  License start month (1-12 or Jan-Dec): 12
  License start day of the month (1-31): 31

License expiration information -
  Expiration year (2003 - 2130) or <Enter> for no
  expiration: 2003
  Expiration month (1-12 or Jan-Dec): 6
  Expiration day of the month (1-30): 1

Should Commuter licenses be issued?(Y/N)[Y]: n

Max number of concurrent tokens [0 - 32766] or <Enter> for no
limit:30

Soft limit [0 - 30] or <Enter> for no limit: 25

Application-server locking mode -
  [0] - Server locked
  [1] - Client&Server locked
  [2] - Unlocked
  [3] - Client locked
  Please enter choice: 1

Lock to multiple redundant servers? (Y/N)[Y]: n
```

```
Locking information (primary or group 1) for server -

 Lock selector for this machine in hexadecimal (if known)
 <Enter> to skip:

 ID PROM (Unix workstation hostid) in hexadecimal
 <Enter> to skip: 0xB345

 IP Address (such as 204.143.137.220)
 <Enter> to skip:

 Disk ID in hexadecimal
 <Enter> to skip:

 Host Name
 <Enter> to skip:

 Ethernet Address (such as 00-23-A4-B8-92-FE)
 <Enter> to skip:

 NetWare IPX Address in hexadecimal
 <Enter> to skip:

 NetWare Serial Number in hexadecimal
 <Enter> to skip:

 Sentinel Computer ID
 <Enter> to skip:

 Custom Host ID in hexadecimal
 <Enter> to skip:

 CPU Serial No in hexadecimal
 <Enter> to skip:

 Note:Lock selector for this machine is 0x1
      Locking code for this machine is 0x132B
Locking information (secondary or group 2) for server- Lock
selector for this machine in hexadecimal (if known)

<Enter> to skip: 0x2

Lock code for this machine in hexadecimal (if known)
<Enter> to skip: 0x27226
```

```
Do you wish to node-lock the clients only to the server host?
(Y/N) y

Should licenses be issued if clock tampering is detected? (Y/
N): n

Available normal license count : 64115

License count that will be decremented from license meter - 6

Do you want to generate license? (Y/N) :y

Type of license string generated -
  [0] - Encrypted license string
  [1] - Expanded readable license string
  [2] - Concise readable license string
  Please enter choice : 0
Please Wait, Generating License...

Message (VLS0041): Your Sentinel LM license meter(s) have been
decremented by 6 units. You now have 64109 units left out of
an initial count of 65500 units.

Do you want to generate more licenses? (Y/N): n
```

Note: Since some features are mutually exclusive, not all options will always be available. For example, in the case of redundant licenses, the command-line mode will ask for primary server-locking information only and will not ask for secondary server-locking information.

Displaying the Base License Code

At times you may need to analyze and display the license code information. To display the code, navigate to *lsdecode.exe* in the *\Delivery\admin.net\Wint32*, *\Delivery\admin.sta\Win32* or *\Tools* subdirectory in which you installed Sentinel LM. The following is a sample license code displayed by lsdecode:

```
  License code :”EL09SHJKF6764873DFU99HKHDKF43HJUYUFD#
  “DECODE” version “1.0”, expires Midnight of Oct 1, 2005,
  exclusive”

 License Type              Normal Networke

 Feature Name              “DECODE”

 Feature Version           “1.0”

 Max concurrent users      2

 Soft limit on users       Unlimited

 License start date        Morning of Oct 1, 2003

 Expiration date           Midnight of Oct 1, 2005

 Additive/exclusive        Exclusive license (overrides
                           additive licenses)

 Held licenses             Held licenses disabled.

 Token lifetime            120 secs (2 min(s))
 (heartbeat)

 Action on clock tamper    No more fresh licenses will
                           be issued

    Press Enter to continue.
```

Note: lsdecode will not display the challenge/response secrets.

Upgrade License Code Generator Using WlscGen

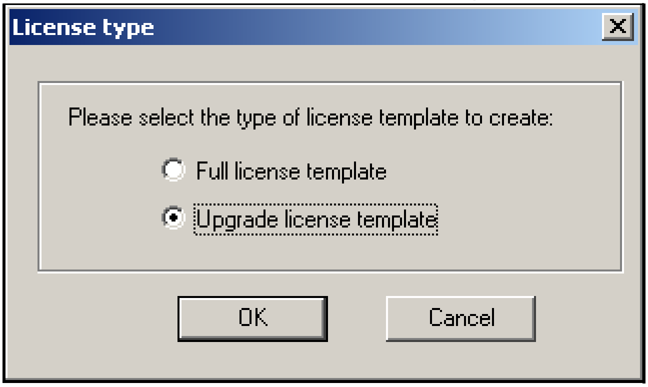
Once you have generated the license code using the **Full license template** option, WlscGen allows you to generate the upgrade license code to upgrade the version and capacity of your base license (license generated by Full license template in WlscGen or by using *lscgen.exe*).

Version upgrade is allowed for non-capacity, pooled and non-pooled base licenses but capacity upgrade is allowed only for non-pooled capacity licenses.

Create upgrade license template using WlscGen

1. Click **New** on the **License Template** screen as shown in Figure “WlscGen License Templates,” on page 91. This opens the License type dialog box.

2. Click on the **Upgrade license template** option on the License type dialog box and then click **OK**. Refer to the figure below.



Please select the type of license template to create:

○ Full license template

● Upgrade license template

License Type Dialog Box

3. You can now access the License features, New template file dialog box for creating new upgrade license template.

   For details on creating a new template or editing an existing one, we suggest you to refer to the Windows License Code Generator Help.

4. After you have created the upgrade license template it gets added to the **License Template** screen as shown in Figure “WlscGen License Templates,” on page 91. On the License Template screen select an upgrade license template and

- ❑ Click **Edit** to edit an existing template file.
- ❑ Click **Remove** to delete an existing template file.
- ❑ Click **Generate** to generate upgrade license code.

To access Windows License Code Generator Help, click **Help Topics** in the **Help** menu or press F1.

The Upgrade License Code Generator Command Line Version - Ulscgen

Upgrade license code can also be generated by the upgrade license code generator command line utility i.e. *ulscgen.exe.*

Note: Before generating the upgrade license code you need to generate the upgrade lock code for the base license using *echouid.exe.* For more details refer to Chapter 12, "Using Upgrade Licensing," on page 187.

Here is a sample of command line upgrade code generator user interface:

```
Sentinel LM 7.3.0 Upgrade License Generation System
    Copyright (C) 2004 Rainbow Technologies, Inc.

Please type the desired responses at the prompts below.

Base Feature Name (any printable characters): f1
Base Feature Version (any printable characters): v1
Upgrade Code :
EKUMEM9ZJDF92U396CFLWE3D6BJGBYWT2SQ8WTCQA3JQ8ENV6ULFP9D
5FAAKQ3C9HX964U6VWP2CQ7O#

Upgrade licensing options -
  [0] - Version upgrade
  [1] - Capacity upgrade
  [2] - All
Please enter choice: 2
Upgrade Version (any printable character): v2
Upgrade License Capacity options -
  [0] From 1 through 1022 (Multiple of 1)
  [1] From 10 through 10220 (Multiple of 10)
  [2] From 100 through 102200 (Multiple of 100)
  [3] From 1000 through 1022000 (Multiple of 1000)
  [4] From 10000 through 10220000 (Multiple of 10000)
  Type an option that suits you best: 1

Type Upgrade Capacity of license or press <Enter> for
unlimited capacity or type 0 for zero capacity: 2000

Available normal license count : 3096

License count that will be decremented from license
meter - 250
Do you want to generate upgrade license? (Y/N) : y

Please Wait, Generating Upgrade License...
Message (VLS0041): Your Sentinel LM license meter(s)
have been decremented by 250 units. You now have 2846
units left out of an initial count of 4096 units.

Do you want to generate more upgrade licenses? (Y/N) : n

Saved new upgrade license code(s) to "upgradelic1".
```

The following table briefly describes the upgrade license code generator command-line options.

Upgrade Code Generator Command Line Options

Option	Description
-h	Displays the help for upgrade license code generator.
-ver	Enables the user to generate an older version library upgrade code.
-upid	Specifies the upgrade lock code of base license.
-E	Specifies the Feature Name of the base license that needs to be upgraded.

Upgrade Code Generator Command Line Options

Option	Description
-v	Specifies the Feature Version of the base license that needs to be upgraded
-u	Bit wise flag. Controls what will be updated: • VLSucg_UPGRADE_VERSION • VLSucg_UPGRADE_CAPACITY • VLSucg_UPGRADE_ALL
-uv	Specifies the new version for this feature.
-cap_units	Determines capacity least count.
-cap	Specifies capacity increment for this feature.

Tip: If the base license is a non-capacity license then the upgrade license code generator shall not ask for capacity upgrade option. The version upgrade option shall be set by default.

Things to Remember with Upgrade Licenses:

As a developer you need to remember the following while generating upgrade licenses:

1. Upgrade licenses cannot be added or deleted dynamically and have to be applied during the server start-up only.

2. Trial licenses, commuter licenses, and redundant licenses cannot be upgraded.

3. Upgrade licenses are available as encrypted licenses only.

4. Pooled capacity licenses cannot be upgraded for capacity.

5. Non-capacity licenses cannot be upgraded for capacity.

Displaying the Upgrade License Code

To verify and display the upgrade license code you need to navigate to *ulsdcod.exe* in the *\Delivery\admin.net\Win32*, *\Delivery\admin.sta\Win32* or *\Tools* subdirectory of the directory in which you installed Sentinel LM. The following is a sample license code displayed by ulsdcod:

```
    Sentinel LM 7.3.0 Upgrade License Decoding Utility

      Copyright (C) 2004 Rainbow Technologies, Inc.

Reading upgrade license codes from file: "C:\Program
Files\Rainbow Technologies\SentinelLM\7.3.0\English\
Delivery\Admin.net\Win32\ulservrc"

Upgrade License code:
"HNB6XQCVHDYHWL6QDH5JYBMNWDXDVF5CQ4VRTCUHHBJMQNLB6TCR2D
ZPVBDS7QCWFQYLPGB927SLEN4BEVMXSMLJHDH847AW2WBWELCXEC#
"f1" version "v1""

License Type                     : Version Upgrade &
                                   Capacity Upgrade

Base License Feature Name        : f1

Base License Feature Version     : v1

Upgrade License Feature Version : v2

Capacity Increment               : 2000

Press Enter to continue . . .
```

Product Distributor License Code Generator

To allow your product distributors to create license codes, you will need to ship the license code generator to them. But first you will need to protect it from unauthorized use. The Distribution Control Program locks the license code generator to a specific Sentinel LM license meter key by writing unique identification information to that hardware key. This ensures that your distributor must buy license meter keys from you, allowing you to track the number of products activated by your distributor.

The Distribution Control Program, *licgensp.exe*, is in the *\Tools* subdirectory of the directory in which you have installed Sentinel LM.

Protecting the License Code Generator

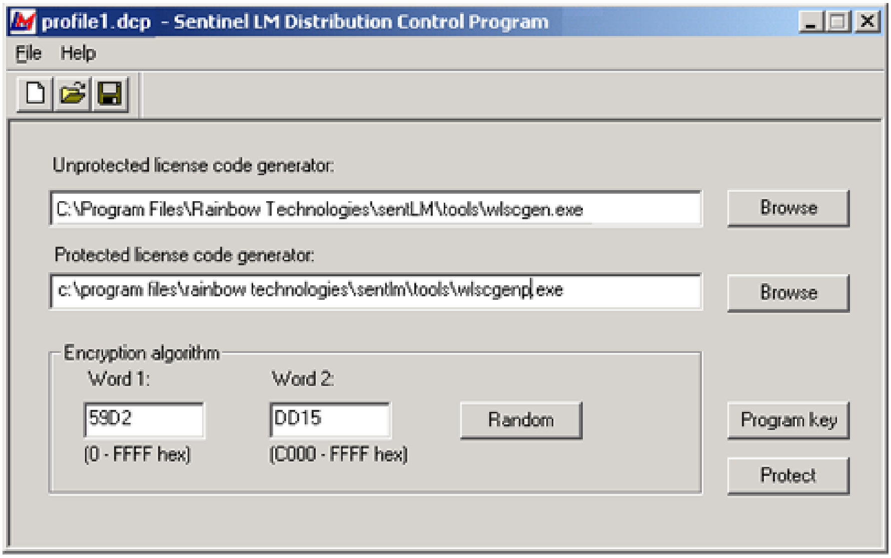
You will need to purchase one license code generator license meter key for each distributor to whom you want to send the protected license code generator.

Here is how to protect the license code generator:

1. Attach the license meter key intended for your distributor to the parallel port of the computer on which you are going to run the Distribution Control program.

2. Click **Start** and select **Run**. Click **Browse** to find the *licgensp.exe* file in the *\Tools\distctrl* subdirectory of the directory in which you installed Sentinel LM. Click **Open** and then click **OK**.

   You now see the Distribution Control Program’s main screen. If you have previously saved a session as a profile file, and want to use those settings again, select **Open** in the **File** menu to select the profile file.



Unprotected license code generator: C:\Program Files\Rainbow Technologies\sentLM\tools\wlscgen.exe

Protected license code generator: c:\program files\rainbow technologies\sentlm\tools\wlscgenp.exe

Encryption algorithm

Word 1: 59D2 (0 - FFFF hex)

Word 2: DD15 (C000 - FFFF hex)

Protecting License Code Generator

3. On the Distribution Control Program’s main screen, in the **Unprotected License Generator** text box, type the location (path) and file name (*WlscGen.exe* or *Lscgen.exe*, depending on whether you want to protect the Windows-interface or command-line version) of the license code generator. Or, click **Browse** to navigate to the location of the existing file, select that file, and click **Open**.

4. In the **Protected License Generator** text box, type the location and filename of the protected version of the license code generator. The Distribution Control Program will create the new file in that location using that name. Or, click **Browse** to navigate to the desired location of the new file.

5. Accept the random values entered into the **Encryption Algorithm** text boxes or type your own values. Or, generate new random values by clicking **Random**. These hexadecimal values will be written to the license meter key to customize it for this version of the protected license code generator.

6. Click **Protect** to protect the license code generator.

⚠ The next step, programming the license meter key, can only be done once per key. After you program the key, you will not able to return it to its original values or re-program with other encryption values.

7. Click **Program Key** to customize the license meter key.
8. To save these settings for future use, select **Save** or **Save As** from the **File** menu to create a profile file.

Tip: It is a good idea to use the **Save** or **Save As** command in the **File** menu to save the current settings as a profile file using a name you will associate with a particular distributor (example, file *Smith100.dcp*) so you can program another key for that distributor in the future that will be compatible with this version of the protected license code generator.

To Create Multiple Customized License Meter Keys

After protecting the license code generator and programming a key as discussed in the previous section, you can remove the license meter key, attach a new one, and click **Program Key** to program an additional license meter key using the same encryption values. (This assumes you want to use the same version of the protected license code generator for multiple distributors.) You can repeat this step to program more keys.

Shipping the Protected License Code Generator

Here is what you need to ship to your distributor:

- A copy of the protected WlscGen or lscgen executable file.
- If you ship *wlscgen.exe*, you also need to ship the *lmlicgen.usr* file, which defines the users of the license code generator and their passwords. You can use WlscGen to define the users and passwords in *lmlicgen.usr* before sending it to your distributor.

- A license code generator Meter key that has been customized by the Distribution Control program for the version of the protected license code generator you are also sending to your distributor.
- Instructions on using the license code generator. An Adobe Acrobat on-line version of the *Sentinel LM Developer's Guide* containing instructions on using WlscGen and lscgen are in the Sentinel LM release. You may also purchase additional printed copies of this manual.
- Copies of the software (and manuals) you want your distributor to ship to the end user.

Chapter 9

Using Redundant License Servers and License Balancing

Redundant license servers and license balancing are of particular interest to your customers who have large network installations and who are concerned with one or more of the following issues:

- Allowing your customers to acquire license tokens even if a particular license server goes down (license server backup).
- Optimizing the speed with which a customer acquires a license token by distributing tokens among multiple license servers to reduce the traffic for a particular license server (license balancing).

  One important aspect of this feature is that, while the network administrator can allocate license tokens among the license servers as he or she desires, Sentinel LM will, over time, reallocate tokens to reflect their actual use. This means that optimal token balancing is done automatically without any intervention on the part of the administrator.
- Accommodating license management with fast response time on wide-area networks that span large geographical areas.

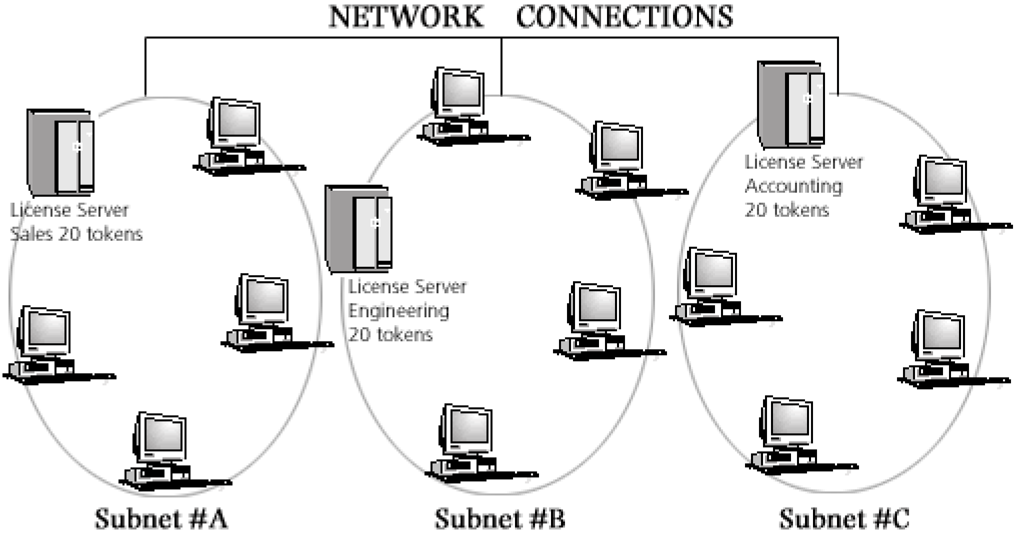
Setting up and using redundant license servers and license balancing require actions on your part, the developer, as well as your customer. We will discuss the details of using redundant license servers and license balancing later in this chapter.

Overview of License Balancing and Redundant License Servers

The method by which Sentinel LM implements license balancing and license server backup is by using redundant license servers. All of the redundant license servers at your customer’s 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. Your customer has a license code that permits 60 instances of a word processing application to run. (We say that the license code authorizes 60 users or that it has 60 *tokens*.)

Your customer wants back up license servers so that if one goes down, all existing attached clients will automatically switch over to another license server and will still be able to acquire license tokens. The diagram below shows one possible configuration. Your customer’s network administrator selects three computers on the network and installs 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 token acquisition.)



Total 60 license tokens in the pool

Sample Configuration of Redundant License Servers and Clients

Note: 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.

The network administrator decides 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 license tokens than the others, that group's license server will borrow tokens from the other license servers automatically. This improves performance because users will be less likely to wait for a token 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.

The network administrator 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.

Note: The following sections mention the rlftool, Wrlftool, WlmAdmin, and lspool utilities. These are utilities your customers will use at their sites to create and maintain redundant license server pools. Because these utilities are intended for the use of your customers, they are documented in the *Sentinel LM System Administrator’s Online Guide*, which you may distribute to your customers.

The Redundant License File (*lservrlf*)

The *redundant license file* (by default named *lservrlf*) is the key to the redundant license server pool. This file is created by the rlftool or Wrlftool utility at your customer’s site, and defines which license servers make up the redundant license server pool, and 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.

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 explain a little more about 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, 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 to be started up. However, Wrlftool 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. To guarantee that a certain license server becomes the leader, use Wrlftool/rlftool to give that license server a preference number of 1 and start it up 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 token 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.

A pool server can exist in 3 states for any license:

- Active
- Inactive
- Unknown

Consider a scenario where you have a pool of 5 redundant servers. The active servers are the ones that are granting license request of the clients. In this case if you add a server locked redundant license and in case the locking criteria does not match with any server then even though the license will be added on the server but the server will not be able to grant any license request. Such a server is marked as inactive for the license.

In case the leader server is unable to determine the status of any server then that server is marked as UNKNOWN.

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

Majority Rule

While creating a redundant license code you may require that its 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 active 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. For more details see “The Redundant License Server Leader” on page 140.

What is the purpose of majority rule? If majority rule is not in effect, your customer could set up the redundant license servers and then separate them on different networks. Each redundant license server, not being able to see any of the others, would then deliver the total number of tokens that should have been distributed among all of the redundant license servers. This would allow your customer to double, triple, quadruple, etc., the total license count without purchasing that many license tokens. So, although you can turn majority rule off, you incur a huge security risk in doing so.

Note: We recommend that you always enable majority rule (the default for WlscGen and lscgen).

The Redundant License Free Pool

In addition to distributing license tokens among redundant license servers, Sentinel LM 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 your customer does 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 transferred from the free pool to the redundant license server requesting tokens. Or, the administrator can use WlmAdmin or 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 single redundant license server goes down, the license tokens it was allocated are handled by the other redundant license servers, and the end user now acquires authorizations 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 142) 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.)

Tips and Recommendations

- The servers in the redundant license server pool must have the same version of Sentinel LM.
- Because license queues are attached to a particular license server, if a redundant license server containing a queue goes down, that queue will not be transferred to the other redundant license servers, and requests in the queue will be terminated.

- Each computer on which a redundant license server resides must have a static, fixed IP or IPX address. Therefore, do not use DHCP to dynamically allocate IP addresses for those computers.
- A server can exists only in one license pool.
- Be sure your customers know that setting up redundant license servers requires the customer to have access privileges to all computers and areas of the network on which license servers will be installed. If your customer makes changes to the redundant license file, those changes will not be transferred to all license servers in the redundant license server pool unless your customer has network and write access to each of the redundant license server computers. If any of these computers run Windows NT/2000/XP/2003, the customer must have administrator privileges to make changes that affect the license servers on those computers.
- If majority rule is in effect, and over half of the redundant license servers go down, the remaining redundant license servers will not service requests for the license that was configured to use majority rule (although non-redundant license codes will continue to be serviced). You will need to include extra code in your application to exit your customers gracefully from this situation.

Enabling/Disabling Automatic License Balancing

The most important step for setting up license balancing is to set up redundant license servers as discussed above. Because Sentinel LM automatically determines the optimal license balance among the redundant license servers, your customer actually does not need to do anything else to balance licenses.

When license balancing is in effect, regardless of how license tokens are initially distributed among the redundant license servers, Sentinel LM will, over time, redistribute them based on actual use. This provides the optimal distribution of license tokens for network users. The way license balancing is achieved is that when one redundant license server is running low on license tokens (based on a configurable borrowing threshold value), it *borrows* tokens from another redundant license server.

The Wrlftool (for Windows) or rlftool (for DOS or UNIX) utility is used by your customer to define the *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.

WlmAdmin or lspool are used to turn borrowing off for a particular license code feature/version or all feature/versions.

Tip: From within WlmAdmin, you can use the **Edit > Redundant License File** to call Wrlftool.

Note: If your customer wants to use license balancing, at least one license token must be allocated to each redundant license server.

Your customer may choose to turn license balancing off by disabling license borrowing or by assigning all tokens to a single redundant license server (zeroing the rest of the license servers). More detailed instructions are included for your customer in the *Sentinel LM System Administrator's Online Guide*.

Commuter Licensing

A commuter license enables your customer to “check out” a temporary authorization to use a license on a portable computer. (See Chapter 10, “Using Commuter Licensing,” on page 155 for general information on using commuter licenses.) You select whether a network license code will be a commuter license when you create 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.

Each time a commuter authorization is checked out, the available number of tokens decrements by one; when the authorization is checked back in or when it expires, the number of tokens increases by one.

Your customers can use Wcommute (for Windows) or lcommute (DOS 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 less 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.

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 the reservation file editor, Wlsgrmgr, see the *Sentinel LM System Administrator's Online Guide*.

Generating License Codes - The Developer's Role

The developer's part of enabling the customer to use redundant license servers is to provide license codes configured to be used with redundant license servers.

Both WlscGen and lscgen allow you to define that a license code will be used on redundant license servers. Whichever tool you use, you will need to obtain the locking codes of all computers on which redundant license servers will reside if you want to lock the license servers to specific computers. We strongly recommend that you do so to increase security for your license codes.

We recommend that at the same time, you ask your customers for the host names of the computers and record that information with the license code you create (within WlscGen, you can use the **Log comment** or **Sales order number** text boxes) so that in the future if the customer forgets which computers contained the redundant license servers for that license code, you can remind them.

Again, to increase security, when you create the license code we strongly recommend that you specify majority rule. For information on why you want to use majority rule, see “Majority Rule” on page 142.

If you specify server locking, you must provide locking codes for all the redundant license server computers.

Redundant License Characteristics

A redundant license code:

- Must be a network license code.
- Can not be a short, numeric license code.
- May enable license sharing. If license sharing is enabled then all the users sharing a token(as per the sharing criteria) must obtain the shared tokens from the same redundant license server.
- May specify holding. If license sharing is enabled then all the users sharing a token(as per the sharing criteria) must obtain the shared tokens from the same redundant license server.
- May be a commuter license (see “Commuter Licensing” on page 145). Notice, once a license authorization has been checked out, it must be checked in to the same redundant license server from which it was checked out.
- May be exclusive.

- May be additive. If it is, 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. For details on these differences, see Chapter 8, “Details on License Code Generator Choices (for WlscGen and lscgen),” on page 102.

Getting Locking Codes from Your Customer

If you want to lock the redundant license servers to specific computers, you will need to tell your customers how to use echoid to get locking codes for each of those computers. You will then specify the locking codes when you create the license code that will be served by the redundant license servers.

More detailed instructions on getting locking codes are included for your customer in the *Sentinel LM System Administartor’s Online Guide*.

Setting up Redundant License Servers - Your Customer’s Role

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

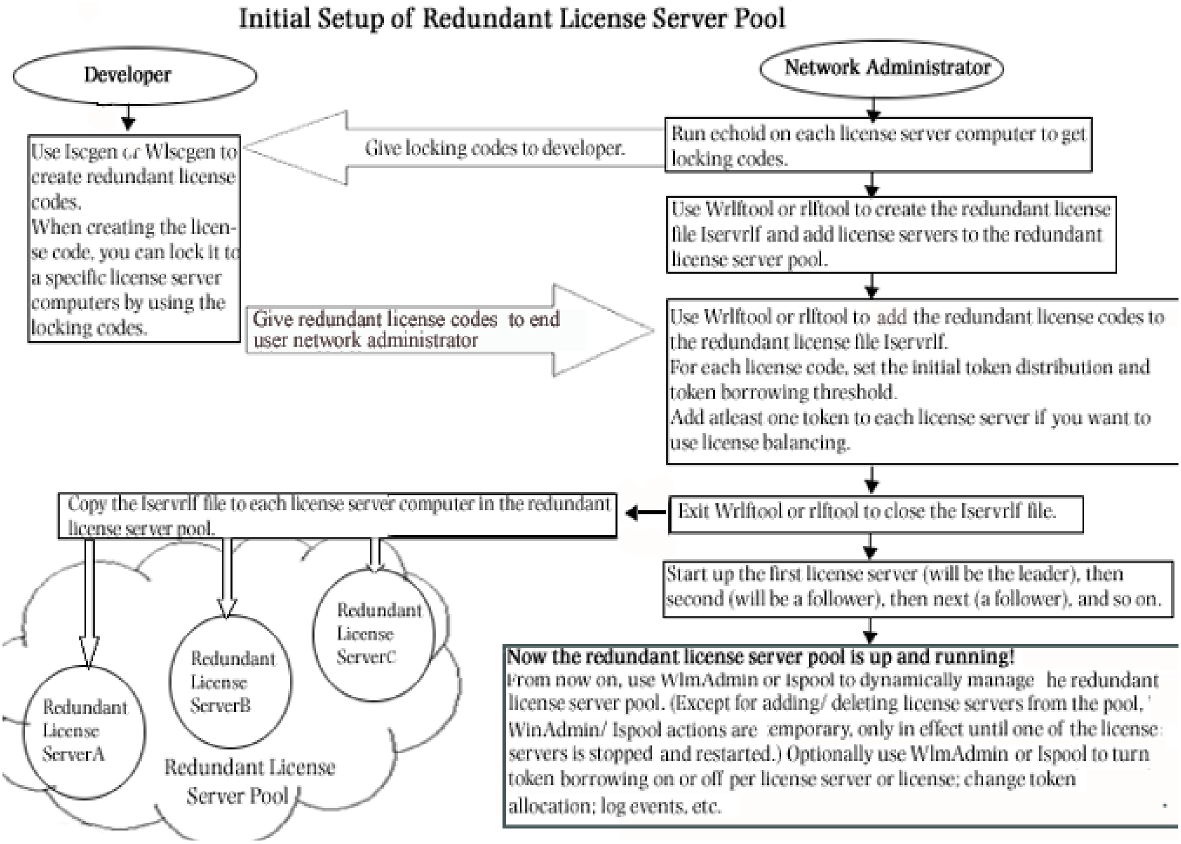
1. First, your customer must decide how many redundant license servers to set up and select the computers on which they will reside. Various factors, including network performance, affect this decision. If you want the license servers to be locked to specific computers, the customer must give you the locking code for each computer on which a redundant license server will be installed.

2. The license codes you ship to the customer who wants to use redundant license servers must identify that license code as one that will be used on a redundant license server. You do this when you

define the license code using WlscGen or lscgen. When defining the license code, if the license server will be locked to a particular computer, you enter the locking codes for each computer on which a redundant license server will reside.

3. Customers must set up the redundant license servers at their site, and create a redundant license file by using Wrlftool/rlftool. After the redundant license server pool is set up, the customer uses the standard Sentinel LM utilities (such as WlmAdmin or lsmon) to monitor the redundant license server pool. The lspool and WlmAdmin utilities allow your customer to dynamically reconfigure the redundant license server pool and set license token allocation.

The following diagram gives an overview of how to initially set up a redundant license server pool. For more information, see the following sections.


Initial Setup of Redundant License Server Pool
Developer
Network Administrator
Use lscgen or Wlscgen to create redundant license codes.
When creating the license code, you can lock it to a specific license server computers by using the locking codes.
Give locking codes to developer.
Run echoid on each license server computer to get locking codes.
Use Wrlftool or rlftool to create the redundant license file lservrlf and add license servers to the redundant license server pool.
Give redundant license codes to end user network administrator
Use Wrlftool or rlftool to add the redundant license codes to the redundant license file lservrlf.
For each license code, set the initial token distribution and token borrowing threshold.
Add atleast one token to each license server if you want to use license balancing.
Copy the lservrlf file to each license server computer in the redundant license server pool.
Exit Wrlftool or rlftool to close the lservrlf file.
Start up the first license server (will be the leader), then second (will be a follower), then next (a follower), and so on.
Redundant License ServerA
Redundant License ServerB
Redundant License ServerC
Redundant License Server Pool
Now the redundant license server pool is up and running!
From now on, use WlmAdmin or lspool to dynamically manage he redundant license server pool. (Except for adding/ deleting license servers from the pool, WlnAdmin/ lspool actions are emporary, only in effect until one of the license servers is stopped and restarted.) Optionally use WlmAdmin or lspool to turn token borrowing on or off per license server or license; change token allocation; log events, etc.

1. First the customer installs (but doesn’t run) the license servers on different computers on the network.

Note: These license servers must be the same version, at least Sentinel LM version 7.0.

2. Next the customer sets 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, the customer can choose not to set this environment variable and allow

the client to broadcast for an available license server. Or, you, the developer, can use the VLSdiscoverExt API call in your application to get information on the license servers in the pool and then choose which license servers to use.)

3. If you, the developer, want the license servers locked to specific computers, you will give the customer instructions on how to run echoid on each computer and record the locking code. The customer should also record the host name of each computer and its IP or IPX address. The customer will give the locking codes, host names, and addresses to you so that you can create the license codes for these license servers.

4. If any license servers are running at the customer site that will be used as redundant license servers, the customer must shut them down before using rlftool or Wrlftool to set up redundancy.

5. The customer will now install the protected applications on the end user computers.

6. The customer gives the developer the locking information (and the host names), and receives the license code(s) that will be served by the redundant license servers.

7. The customer uses Wrlftool or rlftool 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.

Note: All license servers in the pool must use either IP *or* IPX, but you cannot mix protocols. This defines which license servers are in the redundant license server pool.

   ❑ Set the preference number of the license servers. The first license server added to the pool and started up first is known as the leader and all other license servers are known as the followers. The preference order specifies the order in which the leader will be chosen if license servers go down.

8. The customer uses Wrlftool or rlftool (or the WlmAdmin **- Edit > Redundant License File** command, which calls Wrlftool) to add the redundant license codes received from the developer into the redundant license file and set their initial token distribution among the redundant license servers.

9. Since this is the first time that the redundant license server pool has been set up, the customer copies 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 but only when one of the license servers is stopped and restarted.).

10. The customer brings up each of the license servers in the pool. (See the *Sentinel LM System Administrator's Online Guide* for details on license server startup.)

11. The customer 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, the customer can use lspool or WlmAdmin to change license token distribution, view information about the redundant license server pool, turn borrowing on/off, and so on. The customer may use these utilities from any computer on the network but must set the LHOST environment variable to point to one of the redundant license servers. In fact, we strongly recommend that the LSHOST environment variable be set to point to the computer containing the *leader* redundant license server.

Most lspool options dynamically change the redundant license server pool configuration, but do not write the changes permanently to the redundant license file. When the redundant license servers are restarted, the changes are lost. The lspool options that write to the redundant license file are adding and deleting a license server to or from the pool.

To make permanent changes to the redundant license file, your customer can use Wrlftool (or call Wrlftool to make the changes by using the WlmAdmin **- Edit > Redundant License File** command).

Finally, the customer can use lslic to dynamically or permanently add a license code to the redundant license server pool.

More detailed information on WlmAdmin, lspool, lslic, Wrlftool, and rlftool are included for your customer in the *Sentinel LM System Administrator's Online Guide*.

Adding a Redundant License to an Existing Server Pool

The lslic utility can be used to add a redundant license code dynamically by installing it in the license server's memory (to be in effect only until all redundant license servers are stopped and restarted again). If the requested license server is a follower, it will forward the request to the leader; the leader server will then check the validity of the license. If the license is valid, the leader will add it and forward the license code to all the followers.

lslic can also be used to install a redundant license code both in license server memory and permanently in the redundant license file. In this case, the license code is in effect even after the license servers are stopped and restarted.

If distribution criterion is not specified using lslic, the server will equally distribute the tokens among the redundant license servers.

For more information on lslic, see Chapter 13, "Utilities," on page 193.

Chapter 10

Using Commuter Licensing

Commuter licensing allows your customer 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 network user needs to travel on business and wants to take along the protected application. If the license code for that application has been defined by you 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. A license can also be checked out for a computer that does not have direct access to the server. In such a case any system that can access the server can check out the license for a remote system by using remote checkout.

To check out an authorization, and to check it back in later, the network user uses the Wcommute or lcommute utility. Wcommute is a Windows-interface utility; lcommute is a command-line utility for UNIX or Windows (used in a DOS command window).

When checking out an authorization, the user specifies the lifetime (up to a maximum of 30 days) of the authorization. When the authorization expires, the protected application can no longer be used on that laptop. If the user does not check the authorization back in, the license token used by the authorization will not become available again on the network until the authorization expires.

Locking the Authorization

No actual license is transferred to the portable computer. However, the authorization is locked to the portable computer (using the hard disk ID for Windows computers or the Host ID for UNIX computers). If for some reason you prefer to lock to other fingerprint elements (for example, you prefer to use a Computer ID key), you will need to provide a *commute.dat* file to your customer. This file has the same format as the *echoid.dat* file and tells Sentinel LM what to lock to in the case of a commuter authorization. The *commute.dat* file must reside in the same directory as the Wcommute or lcommute utility. For information on the *echoid.dat* file format, see Chapter 7, “How to Select Fingerprint Criteria,” on page 74.

Note: Commuter license authorizations are locked to the portable computer without regard to whatever locking criteria might have been included in the original license code that you generated. This allows a single license code to be used both as a network license, which can only be run on certain computers connected to the network, and a commuter license that can be used on any portable computer that is not connected to the network.

Although no license is transferred to the portable computer, the tokens for the license on the license server decrement by one for each commuter authorization checked out and then one token is released for use when that authorization is checked back in or it expires.

Sentinel LM Application Library Commuter Licensing Calls

As a developer, you can provide extra convenience for your customers, letting them bypass using Wcommute and lcommute, by including the Sentinel LM Application Library commuter licensing calls in your application to:

- Include commuter authorization check-out/check-in commands on the menu of your application;
- Provide your own, customized, stand-alone utility for handling commuter license authorizations.

The API calls provided are:

API Call	Description
VLSgetCommuterInfo	Return commuter license information.
VLSgetAndInstall CommuterCode	Obtain the commuter license authorization from the license server and issues it to the client.
VLSuninstallAndReturn CommuterCode	Remove the commuter license authorization from the client and release the license token on the license server.

For details, see the *Sentinel LM Programmer's Reference Manual*.

Your Protected Application

To work best with commuter licensing, we recommend your application be linked with the dual library for network and stand-alone licensing. (Or, if using Sentinel LM-Shell, protect the application for both stand-alone and network use.) At the least, your application must be able to operate in stand-alone mode when the computer is disconnected from the network.

If you are calling the Sentinel LM Application Library to have your application check out a commuter authorization, then your application *must* be built with the network/stand-alone dual library.

Note: The portable computer must not be configured to use a specific license server. This means, do not set the LSHOST or LSFORCEHOST environment variables (or an *LSHOST/lshost* file) to a network license server on the portable computer. Either these environment variables or files should not be set at all or should be set to no-net to run the application in stand-alone mode.

Creating a Commuter License Code

A commuter license code is created in the same way as other license codes by using WlscGen or lscgen. To create a license code that can be used for commuters:

- Create a network license code.
- Select the **Allow network license to commute** check box when defining the license code template (for WlscGen) or use the -O option when using lscgen or enter 'Y' when asked for "Should commuter licences be issued".
- Remember that this license code may also be used on the network by non-commuters, so set it up to use whatever locking mechanisms or restrictions you would normally specify for a standard network license code. (This locking info will not be used when an authorization to use the license is checked out by a commuter.)

Instructions for Your Customers

These sections contain some information your customers should be aware of when using commuter licensing. These topics are also discussed in the *Sentinel LM System Administrator's Online Guide*.

Restricting Commuter Licensing

Commuter licensing uses the same license tokens as other types of licensing. If your customer wants to restrict the percentage of license tokens on a specific license server that can be used by commuter licensing authorizations, the -com option must be used when starting the license server to specify the percentage of license tokens on that license server that may be used by commuters. 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.

Before Checking Authorizations In/Out

Before the network user checks a commuter authorization in or out, the user's portable computer must:

- (For checkout.) Have the protected application installed and ready to use. Remember that the application must be able to run off the network.
- Have network access to the appropriate license server. (Your customers do not need to set the *lshost* file or LSHOST or LSFORCEHOST environment variables to contact a specific license server because they will be able to request a specific license server when using the commuter licensing utility.). Through remote checkout some other machine connected to the server can check out the authorization for the portable machine to be taken away.
- Be connected to the network.
- Have access to the Wcommute or lcommute utility (either on the network or on the portable computer). Wcommute is a Windows-interface utility; lcommute is a command-line utility for UNIX or

Windows (used in an MS-DOS command window). If you are not locking the authorization to the default, you will need to supply a configured *commute.dat* file which must be in the same directory as Wcommute or lcommute. For information on *commute.dat*, see “Locking the Authorization” on page 156.

Note: A user checking a commuter authorization back into the network *must* check it into the same license server from which it was checked out.

Using the Commuter Licensing Utilities

Your customers will find using Wcommute makes checking a commuter authorization in or out quick and easy. A simple display lists the commuter licenses available on the license servers of interest, and your customer picks the one he or she wants. Information for your customers on how to use Wcommute and lcommute is contained in the *Sentinel LM System Administrator’s Online Guide,* which you may send to your customers.

Your customer will not be able to check out an authorization if one is already checked out on that computer for the same feature/version and locking criteria.

Remote Commuter Licensing

WCommute (the Windows-interface utility) and lcommute (the Windows/UNIX command-line utility) allows a network client to check out a commuter authorization and send it to a remote user.

New utilities, WRCommute (the Windows-interface utility) and rcommute (the Windows/UNIX command-line utility) are now provided to allow the remote user to generate a commuter locking code to send to the local user checking out the commuter authorization from the Sentinel LM license server, and also to install the commuter authorization received from the network user.

A remote user runs WRCommute or rcommute to generate a commuter locking code for the remote computer, and sends the commuter locking code to the local user who has access to the Sentinel LM license server.

The local user runs WCommute or lcommute (entering the remote user's commuter locking code) to check out a remote commuter authorization, and sends it to the remote user.

The remote user runs WRCommute or rcommute to install the remote commuter authorization. The entire process of obtaining and installing remote commuter authorization has been discussed in the following steps:

Step 1: Getting a Remote Commuter Locking Code

The remote user must generate a commuter locking code for his or her remote computer and send it to the network user.

Note: The commuter locking code used to lock a commuter authorization is not the same as the locking code displayed by echoid. You must use WRCommute or rcommute to obtain it.

Getting a Commuter Locking Code with WRCommute

Run WRCommute and click the **Get Locking Code** tab. You see the commuter locking code for the current computer.

📋	To copy the commuter locking code to the computer clipboard, click **Copy to Clipboard**.
💾	To save the commuter locking code to a file, click the **Save to file** button.
🔍	To display the commuter locking code in a separate window, click **Display**.

Tip: To send the commuter locking code as an email message, type an email address in the Email address text box and click **Send**.

Getting a Commuter Locking Code with rcommute

To get a commuter locking code, use the command-line utility rcommute in a command-line window. Rcommute accepts two options:

Option	Description
-l	Get the commuter locking code and store it in the specified file.
filename	Name and path of the file in which to write the commuter locking code.

For example:

```
rcommute -l lock.txt
```

Step 2: Checking Out a Remote Commuter Authorization

The next step is to use WCommute or lcommute on a computer connected to the network containing the Sentinel LM license server to check out a commuter authorization for the remote user.

Checking out a Remote Authorization with WCommute

1. Run **WCommute** and select the commuter authorization you want to use from the network as you normally would.

2. Select the **Check out authorization** for remote machine check box and click **Check Out**. Now WCommute displays the **Locking Code for Remote Machine** dialog box.

   To type the commuter locking code received from the remote user, select **Enter the locking code for remote machine** and type it in the text box.

Tip: To paste the commuter locking code from the computer clipboard, click

Paste from Clipboard.

To choose the commuter locking code from a file, select **Get locking code for remote machine** from file and type the file name in the text box or use the **Browse** button to find it.

3. Click **OK** to access the Remote Commuter Licensing dialog box.

   To send the commuter authorization as an email message, type an email address in the Email address text box and click **Send**.

Checking out a Remote Authorization with lcommute

To check out a commuter authorization for a remote user, use the command-line utility lcommute in a command-line window.

Step 3: Installing a Remote Commuter Authorization

Once the commuter authorization has been checked out for the remote user and sent to him or her, the remote user needs to install it.

Note: A remote commuter authorization cannot be checked back into the license server, but will simply expire on the remote computer.

Installing a Remote Commuter Authorization with WRCommute

4. Run WRCommute and click the **Install Remote Authorization Code** tab.

   - ❑ To type the authorization code received from a network user, select **Enter remote authorization code** and type it in the text box.
   - ❑ To install the authorization code from a file, select **Get remote authorization code(s)** from file and type the file name in the text box or use the **Browse** button to find it.
   - ❑ Click **Install**.

Installing a Remote Commuter Authorization with rcommute

To install a remote commuter authorization, use the command-line utility rcommute from a command-line window. rcommute accepts two options:

Option	Description
-i	Install the commuter authorization from the specified file.
filename	Name and path of the file that contains the remote commuter authorization.

For example:

```
rcommute -i commuter.txt
```

Chapter 11

Using Capacity Licensing

This chapter introduces you to the concept of capacity licensing and discusses how your customer will use it.

First, let's review basic licensing concepts introduced earlier in this guide.

- **Hard Limit** – When you create a license code using the WlscGen or lscgen license code generator, you specify how many copies of that license can be used at one time. This is known as the *hard limit*, and usually reflects the number of different clients who can use the license at the same time. You can also think of this as the number of computers that can run the licensed software at one time.
- **License Token**– Each copy of the license permitted by the hard limit is known as a *token*. Each client normally uses one token of the license.
- **Sharing** – Each token can be used by multiple clients if *sharing* is enabled. The group of clients that can share a token are identified by some common element, such as the same user ID. Sharing is often used to allow multiple clients(having a common user name or host name or X-Display name or vendor defined name) to run the application without consuming more than one token.

Developers find that most of their licensing needs can be satisfied by using the hard limit and sharing concepts above.

However, if you want to create more complex licensing models, capacity licensing provides an additional layer to your licensing by building on these concepts. Capacity licensing expands on hard limit and sharing by allowing you to specify the number of operations a token user can perform. Capacity licensing also allows the end user to use an extra flag along with hard limit to control numer of applications that can run simultaneously.

See the following sections for examples of using capacity licensing and for more information on capacity licensing concepts.

Capacity Licensing Example

Although we are used to thinking of licensing as a way of restricting the number of users who can use a software product, it can also restrict hardware devices or machines controlled by software.

To help us understand the following example of using capacity licensing to control the number of operations performed by a machine, consider how cell telephones work:

- Because a limited number of frequencies exist, cell phones share common frequencies. Cell phone service providers split their service areas into cells, which allows phones in different cells to use the same frequencies at the same time.
- Each service area contains an antenna station that services many cells in that area. As a caller moves out of range of an antenna, that antenna station transfers the caller to another antenna station in another area. Therefore, the number of cells used by an antenna station is dynamic, changing constantly.

Let us consider a company named ZoneCast, which sells equipment used by antenna stations. Let's call their product CellManager. They need to restrict the number of antenna stations that can use CellManagers (the hard limit), but also want to restrict the number of cells that can be serviced by CellManagers (the capacity) for a single antenna station. Although each CellManager may have the ability to service 10,000 cells, not every antenna station needs to service that many cells. Therefore, ZoneCast uses Sentinel LM to create licenses that allows ZoneCast to:

1. Charge by number of antenna stations that use CellManagers (the hard limit), *and;*

2. Charge by number of cells (capacity) the CellManagers used by a single antenna station can service. (You can also share the capacity among multiple antenna stations by using *pooled* capacities—see the next section for details.)

ZoneCast makes just one CellManager, but they can use capacity licensing to charge different amounts for the same CellManager by restricting the number of cells that equipment can service.

Capacity licensing benefits the customer who can purchase more cell coverage by simply purchasing an upgrade license instead of buying and installing an additional CellManager.

Because the antenna stations are networked, the CellManager licensing is managed by a central Sentinel LM license server, which makes licensing maintenance and upgrades easier.

Additional Capacity Licensing Concepts

In addition to the idea of capacity, capacity licensing also introduces the concept of *teams* (similar to the concept of sharing - see Chapter 3, “Planning Your Application Licensing,” on page 35) and *pooled* and *non-pooled* capacities.

A group of users who will share a license token is called a team when you are using capacity licensing. Using our example above, each antenna station that uses multiple CellManagers is a team and the CellManager devices are team members. As we discussed above, the cell phone provider buys a license that specifies the number of antenna stations that can use CellManagers (hard limit or the number of teams) and the number of cells all CellManagers used by an antenna station can service (capacity). The license also specifies the number of CellManagers each antenna station can use (the team limit). The *team limit* specifies the number of members of a team.

If a capacity license is non-pooled, the capacity specified is shared only among the members of a team, but not between teams. Pooled capacity permits all teams to share the license capacity.

To summarize, using our example:

- **Hard limit** - Number of antenna stations (number of teams).
- **Capacity** - Number of cells that can be serviced by all CellManagers connected to a single antenna station (non-pooled capacity) or all antenna stations (pooled capacity). That is, number of operations that can be performed by a member of a team.
- **Team** - A single antenna station to which multiple CellManagers are connected. (The team uses one license token.)
- **Team Size** - Number of CellManagers connected to an antenna station (number of members of a team).
- **Pooled Capacity** - Number of cells that CellManagers connected to all antenna stations can manage (capacity is shared among all teams).
- **Non-pooled Capacity** - Number of cells that CellManagers connected to a single antenna station can manage (capacity is shared among the members of a single team).

Tip: A license server cannot contain both pooled and non-pooled capacity licenses for the same feature/version.

To better understand the concepts of capacity licensing lets consider that the following licenses are available on the server:

Lic No.	Feature	Version	Capacity	Hard Limit	Share Limit	Lic Type	Sharing Criteria
L1	F1	V1	NULL	3	-	non-capacity	None
L2	F2	V2	1500	3	3	capacity-non-pooled	User Name
L3	F2	V2	2000	4	3	capacity-non-pooled	Host Name
L4	F2	V2	3000	2	2	capacity-non-pooled	User Name
L5	F3	V3	5000	3	2	capacity-pooled	Vendor - Code

The above licenses can request for a license in following ways:

Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
1	User1, Host1, Xdisplay1 ,Vendor Code1	F1, V1	NULL	NULL	Granted	Shall work in a similar way as for Sentinel LM 7.2.	L1: 1 token L2: - L3: - L4: - L5: -
2	User1, Host1, Xdisplay1 ,Vendor Code1	F1, V1	1000	100	Not Granted	Server shall return an error that the License available is not a capacity license.	L1: 1 token L2: - L3: - L4: - L5: -
3	User1, Host1, Xdisplay1 ,Vendor Code1	F2, V2	1200	250	Granted	Token#1 from L2 license is issued. Even though that the request was made for a capacity of 1200, the whole license capacity of 1500 is issued. Thus, a capacity of 1500 is reserved for the Team "User1" and 250 out of 1500 is given to this client. This is client#1 of Team "User1".	L1: 1 token L2: -1 tokenTeam:User1(1 Client) L3: - L4: - L5: -
4	User1, Host1, Xdisplay1 ,Vendor Code1	F2, V2	1200	300	Granted	A Shared copy of Token#1 from L2 license is issued. This client is given a capacity of 300 out of Team capacity of 1500. This is client#2 of Team "User1". A total of 550 out of Team capacity of 1500 is used.	L1: 1 token L2: -1 tokenTeam:User1(2 Clients) L3: - L4: - L5: -

Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
5	User1, Host1, Xdisplay1 ,Vendor Code1	F2, V2	1200	1000	Denied	The Capacity left for this team is 1500-550=950	L1: 1 token L2: -1 tokenTeam: User1 (2Clients) L3: - L4: - L5: -
6	User1, Host1, Xdisplay1 ,Vendor Code1	F2, V2	1500	400	Granted	A Shared copy of Token#1 from L2 license is issued. This client is given a capacity of 400 out of Team capacity of 1500. This is client#3 of Team "User1" . A total of 950 out of Team capacity of 1500 is used. As the Team User1 was already exists so the Team Capacity request of 1500 made by this client was ignored.	L1: 1 token L2: -1 tokenTeam: User1 (3Clients) L3: - L4: - L5: -
7	User1, Host1, Xdisplay1 ,Vendor Code1	F2, V2	1200	200	Denied	Sharing Limit exhausted for this Team.	L1: 1 token L2: -1 tokenTeam: User1 (3Clients) L3: - L4: - L5: -

Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
8	User2, Host2, Xdisplay2, Vendor Code2	F2, V2	1500	400	Granted	Token#2 from L2 license is issued. Thus, a capacity of 1500 is reserved for the Team "User2" and 400 out of 1500 is given to this client. This is client#1 of Team "User2".	L1: 1 token L2: -2 tokensTeam: User1 (3 Clients), User2 (1 Client) L3: - L4: - L5: -
9	User3, Host3, Xdisplay3,Vendor Code3	F2, V2	1900	400	Granted	Token#1 from L3 license is issued. Thus, a capacity of 2000 is reserved for the Team "Host3" and 400 out of 2000 is given to this client. This is client#1 of Team "Host3".	L1: 1 token L2: -2 tokensTeam: User1 (3 Clients), User2 (1 Client) L3: -1 tokenTeam: Host3(1 Client) L4: - L5: -

Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
10	User4, Host3, Xdisplay3 , Vendor Code3	F2, V2	2000	500	Granted	A Shared copy of Token#1 from L3 license is issued. This client is given a capacity of 500 out of Team capacity of 2000. This is client#2 of Team "Host3" . A total of 900 out of Team capacity of 2000 is used. As the Team Host3 was already exist so the new Team Capacity request of 2000 made by this client was ignored.	L1: 1 token L2: -2 tokensTeam: User1 (3 Clients), User2 (1 Client) L3: -1 tokenTeam: Host3(2 Client) L4: - L5: -
11	User5, Host5, Xdisplay5 ,Vendor Code5	F2, V2	2500	3500	Denied	As the Tem Member Capacity requested is greater than the Team capacity requested, request is denied at client end itself. No Team is created at server end.	L1: 1 token L2: -2 tokensTeam: User1 (3 Clients),User2 (1 Client) L3: -1 tokenTeam: Host3(2 Client) L4: - L5: -

Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
12	User5, Host5, Xdisplay5 ,Vendor Code5	F2, V2	3000	3000	Granted	Token#1 from L4 license is issued. Thus, a capacity of 3000 is reserved for the Team "User5" and 3000 out of 3000 is given to this client. This is client#1 of Team "User5".	L1: 1 token L2: -2 tokensTeam: User1 (3 Clients),User2 (1 Client) L3: -1 tokenTeam: Host3(2 Client) L4: - 1 tokenTeam: User5(1client) L5: -
13	User5, Host5, Xdisplay5 ,Vendor Code5	F2, V2	2500	100	Denied	No capacity available	L1: 1 token L2: -2 tokensTeam: User1 (3Clients), User2 (1 Client) L3: -1 tokenTeam: Host3(2 Client) L4: - 1 tokenTeam: User5(1client) L5: -

Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
14	User5, Host5, Xdisplay5 ,Vendor Code5	F2, V2	2500	0	Granted from L4	A Shared copy of Token#1 from L4 license is issued. This client is given a capacity of 0 out of Team capacity of 3000. This is client#2 of Team "User5" . A total of 3000 out of Team capacity of 3000 is used.	L1: 1 token L2: -2 tokensTeam: User1 (3 Clients), User2 (1client) L3: -1 tokenTeam: Host3(2client) L4: - 1 tokenTeam: User5(2client) L5: -
15	User6, Host6, Xdisplay6 ,Vendor Code6	F2, V2	0	0	Granted from L2	Token#3 from L2 license is issued. Even though that the request was made for a capacity of 0, the whole license capacity of 1500 is issued. Thus, a capacity of 1500 is reserved for the Team "User6" and 0 out of 1500 is given to this client. This is client#3 of Team "User6".	L1: 1 token L2: -3 tokensTeam: User1 (3 Clients), User2 (1 Client)User6 (1 Client) L3: -1 tokenTeam: Host3 (2client) L4: - 1 tokenTeam: User5 (2client) L5: -

Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
16	User7, Host7, Xdisplay7 ,Vendor Code7	F2, V2	500	200	Granted from L3	Token#2 from L3 license is issued. Even though that the request was made for a capacity of 500, the whole license capacity of 2000 is issued. Thus, a capacity of 2000 is reserved for the Team "Host7" and 200 out of 2000 is given to this client. This is client#1 of Team "Host7".	L1: 1 token L2: -3 tokensTeam: User1 (3 Clients), User2 (1 Client)User6 (1 Client) L3: -2 tokenTeam: Host3 (2 Client), Host7 (1 client) L4: - 1 tokenTeam: User5 (2client) L5: -

Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
17	User8, Host8, Xdisplay8 ,Vendor Code8	F2, V2	1500	1500	Granted from L3	Token#3 from L3 license is issued. Even though that the request was made for a capacity of 1500, the whole license capacity of 2000 is issued. Thus, a capacity of 2000 is reserved for the Team "Host8" and 1500 out of 2000 is given to this client. This is client#1 of Team "Host8".	L1: 1 token L2: -3 tokensTeam: User1 (3client), User2(1client) User6 (1client) L3: -3 tokenTeam: Host3 (2 Client), Host7 (1 client), Host8 (1client) L4: - 1 tokenTeam: User5 (2 client) L5: -

Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
18	User1, Host1, Xdisplay1 ,Vendor Code1	F3,V3	1000	500	Granted from L5	Token#1 from L5 license is issued. Thus, a capacity of 1000 is reserved for the Team "VendorCode1" and 500 out of 1000 is given to this client. This is client#1 of Team "VendorCode1".	L1: 1 token L2: -3 tokensTeam: User1 (3 Clients), User2 (1 Client)User6 (1 Client) L3: -3 tokenTeam: Host3 (2 Client), Host7 (1 client), Host8 (1client) L4: - 1 tokenTeam: User5 (2 client) L5:1 tokenTeam: VendorCode1 (1 client)

Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
19	User1, Host1, Xdisplay1 ,Vendor Code1	F3, V3	1200	400	Granted from L5	A Shared copy of Token#1 from L5 license is issued. This client is given a capacity of 400 out of Team capacity of 1000. This is client#2 of Team "VendorCode1" . A total of 900 out of Team capacity of 1000 is used. As the Team VendorCode1 was already exist so the new Team Capacity request of 1200 made by this client was ignored.	L1: 1 token L2: -3 tokensTeam: User1 (3 Clients), User2 (1 Client)User6 (1 Client) L3: -3 tokenTeam: Host3 (2 Client), Host7 (1 client), Host8 (1 client) L4: - 1 tokenTeam: User5 (2 client) L5:1 tokenTeam: VendorCode1 (2 clients)

Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
20	User1, Host1, Xdisplay1 ,Vendor Code1	F3, V3	1000	100	Denied from L5	Sharing Limit exhausted	L1: 1 token L2: -3 tokensTeam: User1 (3 Clients), User2 (1 Client)User6 (1 Client) L3: -3 tokenTeam: Host3 (2 Client), Host7 (1 client), Host8 (1 client) L4: - 1 tokenTeam: User5 (2 client) L5:1 tokenTeam: VendorCode1 (2 clients)

Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
21	User2, Host2, Xdisplay2 ,Vendor Code2	F3, V3	3000	2000	Granted from L5	Token#2 from L5 license is issued. Thus, a capacity of 3000 is reserved for the Team "VendorCode2" and 2000 out of 3000 is given to this client. This is client#1 of Team "VendorCode2".	L1: 1 token L2: -3 tokensTeam: User1 (3 Clients), User2 (1 Client)User6 (1 Client) L3: -3 tokenTeam: Host3 (2 Client), Host7 (1 client), Host8 (1 client) L4: - 1 tokenTeam: User5 (2 client) L5:2 tokenTeam: VendorCode1 (2 clients), VendorCode2 (1 client)

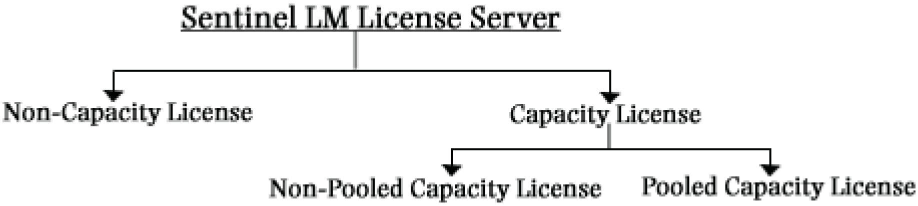
Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
22	User2, Host2, Xdisplay2 ,Vendor Code2	F3,V3	3000	1500	Denied from L5	Team Capacity left out for the Team "VendorCode" is 1000 and the request is made for 1500.	L1: 1 token L2: -3 tokensTeam: User1 (3clients), User2 (1 Client)User6 (1 Client) L3: -3 tokenTeam: Host3 (2 Client), Host7 (1 client), Host8 (1 client) L4: - 1 tokenTeam: User5 (2 client) L5:2 tokenTeam: VendorCode1 (2 clients), VendorCode2 (1 client)
23	User3, Host3, Xdisplay3 ,Vendor Code3	F3, V3	2000	1000	Denied from L5	The left out capacity of the license is 1000 and the Team Capacity requested is 2000	-do-

Req #	Sharing Info	Req. Feature/ ver	Req Team Capacity	Req. Team-Member Capacity	Request Granted from Lic	Result	Lic Server Status
24	User3, Host3, Xdisplay3 ,Vendor Code3	F3, V3	0	0	Granted from L5	Token#3 from L5 license is issued. Thus, a capacity of 0 is reserved for the Team "VendorCode3" and 0 out of 0 is given to this client. This is client#1 of Team "VendorCode3".	L2: -3 tokensTeam: User1 (3 Clients), User2 (1 Client)User6 (1 Client) L3: -3 tokenTeam: Host3 (2 Client), Host7 (1 client), Host8 (1 client) L4: - 1 tokenTeam: User5 (2 client) L5:3 tokenTeam: VendorCode1 (2 clients), VendorCode2 (1 client), VendorCode3 (1 client)
25	User4, Host4, Xdisplay4 , Vendor Code 4	F3, V3	500	100	Denied	Hard Limit Exhausted.	-do-

How to Implement Capacity Licensing

Sentinel LM recognizes three types of licenses: non-capacity, capacity-pooled, and capacity-non-pooled.

The figure below represents the structure of Sentinel LM licensing:



Sentinel LM Licensing

To create a capacity license, you will use the WlscGen or lscgen license code generator to select capacity features. While working with WlscGen, if you select long license code type you can specify the capacity licensing options from the **Licensing Options** dropdown. Further you can define the capacity licensing options in the **Optional License Features** screen.(See Chapter 8, “Creating and Upgrading License Codes,” on page 79 for information on creating license codes. Also, see the WlscGen online help for details on using WlscGen.)

After generating the capacity license you send it to your customer, the customer then uses Wlsgrmgr to set up teams and allocate licenses to teams. For details on using Wlsgrmgr, see the *Sentinel LM System Administrator’s Online Guide*.

Your Customer’s Experience with Capacity Licensing

This section discusses how capacity licenses are handled at your customer’s site, and also how your customer reserves licenses for teams.

License Installation

In the case of non-pooled capacity licenses, Sentinel LM supports multiple licenses for the same feature name and version name but different capacities. For example:

- FeatureX, Version 1.0, Capacity 1000 and Hard Limit 2
- FeatureX, Version 1.0, Capacity 2000 and Hard Limit 4
- FeatureX, Version 1.0, Capacity 1500 and Hard Limit 3

If an additive license with the same capacity but different hard limit for the same feature and version as an existing license is installed by your customer, the hard limits will be merged. If an exclusive non-pooled license with the same capacity but different hard limit for the same feature and version as an additive license is installed by your customer, the additive license will be overwritten if the existing license has the same capacity.

For details on additive and exclusive licenses, see Chapter 3, "Planning Your Application Licensing," on page 35.

Pooled and Non-Pooled Examples

Using our earlier example, let's say that ZoneCast uses non-pooled capacity licenses. The customer buys a license for each antenna station (team) and the license specifies the number of CellManagers connected to each antenna station (team size) and the capacity for all CellManagers (team members) connected to the antenna station.

The customer uses the capacity reservation file to allocate a specific license to an antenna station, depending on the needs of that station.

For example, the customer has five antenna stations:

- Stations 1, 2, and 3 service about the same load and require four CellManagers to service 35,000 cells.
- Station 4 has a much lower load, requiring only one CellManager and 2,000 cells.

- Station 5 has a very high load and requires seven CellManagers and 70,000 cells.

To account for peak times where the usage will be higher than average and also to plan for expected expansion, the customer purchases the following three licenses:

- **License 1** – Feature: CellPhone, Version: 1.0, Capacity: 45,000, Team limit: 5, Hard limit: 3, Non-pooled capacity
- **License 2**– Feature: CellPhone, Version: 1.0, Capacity: 5,000, Team limit: 1, Hard limit: 1, Non-pooled capacity
- **License 3** – Feature: CellPhone, Version: 1.0, Capacity: 80,000, Team limit: 8, Hard limit: 1, Non-pooled capacity

The customer adds two entries to his reservation file:

- Reservation 1: Group A, Team name: "LowUsage_CellPhoneSite", reserves License 2.
- Reservation 2: Group B, Team name: "HighUsage_CellPhoneSite", reserves License 3.

In the case of a pooled license, all teams share the licenses' capacity. Each team can request a different capacity, so long as the sum of all teams' requests does not exceed the total capacity defined in the license. If a team requests a capacity that exceeds the amount available, the request is denied.

For information on using Wlsgrmgr to reserve licenses for teams, see the *Sentinel LM System Administrator's Online Guide*.

Chapter 12

Using Upgrade Licensing

Sentinel LM enables you to upgrade your customer's existing license to change the version or increase the capacity. A special upgrade license must be created to upgrade the existing license.

Your customers benefit from upgrade licensing by being able to use new features available in the latest version of your software.

Implementing Upgrade Licensing

In the current business environment, new versions of software are being released quite often. It is essential to provide free product upgrades for minor releases. Additionally, it is very important to ensure that product upgrades are available to only those customers who all are eligible for them.

Sentinel LM provides two ways by which upgrade licensing is implemented:

- Version Upgrade
- Capacity Upgrade

Version Upgrade

Version upgrades allows you to upgrade the version of existing licenses without increasing the hard-limit of those licenses. For example:

> SecureSoft has recently launched version 3.4 of its product which provides some additional features that were not available in version 3.3. The company wants all clients using version 3.3 to be able to upgrade to version 3.4. By providing its customers with upgrade licenses, SecureSoft ensures that only customers who already have a license for version 3.3 can upgrade to the new version.

Sentinel LM allows end users to upgrade the version of their software by using echouid**.** See "Generating an Upgrade Lock Code" on page 189.

To upgrade to a newer version, the user needs to specify Feature Name, Feature Version, License Type, Lock Info, Time Stamp and Upgraded Version.

Note: Upgrade licenses cannot be generated for Trial Licenses, Commuter Licenses, and Redundant Licenses nor can they be used to extend an expiration date.

The upgrade license feature is applicable only to the base licenses that the customer chooses. Once a base license is upgraded, it cannot be removed/ deleted dynamically. The upgrade license feature needs to be applied during the license server start-up only.

Capacity Upgrade

The capacity upgrade feature of Sentinel LM enables a customer to upgrade the capacity non-pooled licenses. For example, a client using the capacity of 1000 for the team limit of 2 wants to upgrade the capacity to 1200. This can be achieved using the capacity upgrade feature of Sentinel LM.

To upgrade the capacity of the existing license, you need to specify Feature Name, Feature Version, License Type, Lock Info, Time Stamp and Capacity Increment.

For more details on capacity licensing, refer to Chapter 11, "Using Capacity Licensing," on page 165.

Note: Upgrade licenses are available as encrypted licenses only and are not available as readable or numeric licenses.

Instructions for End Users

As a developer, you will generate the license for the upgrade lock code. Additionally you will need to guide your customers through the upgrade licensing process and tell them how to obtain their upgrade lock code.

We recommend you to make your end users aware of the process that they need to adapt to upgrade their licenses:

- To request the license upgrade (version or/and capacity) for a specific product, the end user must call the developer.
- After requesting the upgrade license from the developer, the end user needs to generate the upgrade code for the specific license. The upgrade code is generated with the help of echouid and with support from the developer. For more information on upgrade license code generation, refer to “Generating an Upgrade Lock Code” on page 189.
- Once the upgrade code is generated, it needs to be sent to the developer.
- The developer generates the upgrade license based on the upgrade code and sends it to the end user.

Generating an Upgrade Lock Code

Your customer generates the upgrade lock code and sends it to you, the developer. Use *echouid.exe* to generate an upgrade code. This file is located under *\Rainbow Technologies\SentinelLM\7.3.0\English\Tools.* It is also included in *\Delivery\Admin.net\Win32,*which contains tools you may want to send to your customer.

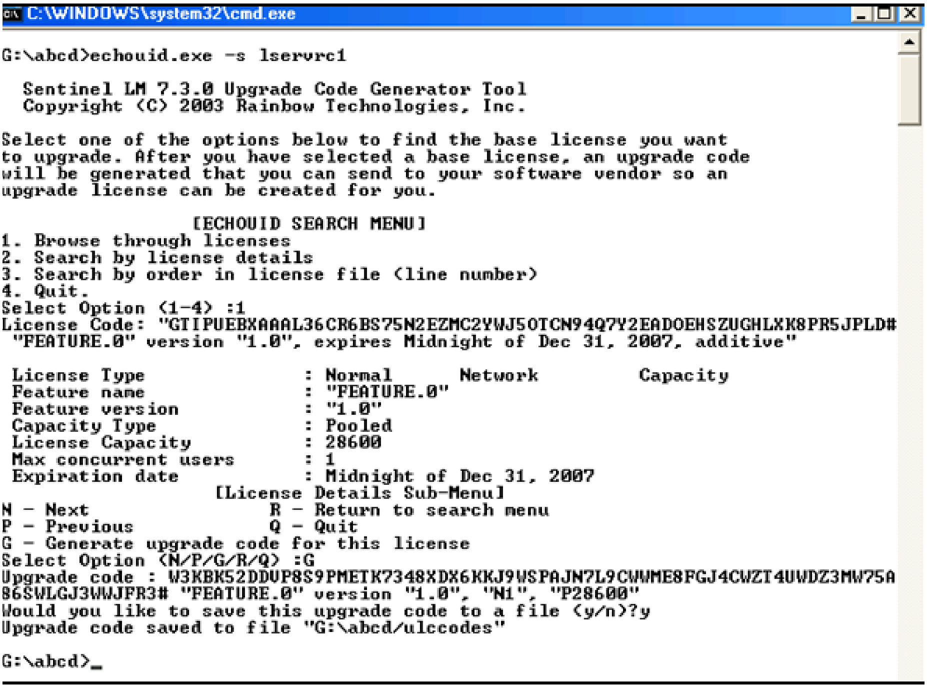
Your customer will open an MS-DOS window to run *echouid.exe* and follow the instructions. The Sentinel LM Upgrade Code Generator Tool searches for *lservrc* in the current directory, so you will want to install it on your customer's computer in the same directory as your application's license file.

Note: Though echouid first searches for *lservrc* but the name of some other license file can also be specified ,either using the switch **-s** or specifying it when asked for.(This is asked if echouid is started without *lservrc* in the same directory or without **-s** switch).

On accessing the **Sentinel LM Upgrade Code Generator Tool** your customer will use the **Echouid Search Menu** to select the base license he or she wants to upgrade. The **Echouid Search Menu** allows you to **Browse through licenses**, **Search by license details** and **Search by order in license file <line number>.**

After selecting the desired base license,your customer will generate the upgrade code from the **License Details Sub-Menu** and then send the upgrade code to you to obtain the upgrade license**.** For details on using echouid, see the *Sentinel LM System Administrator Online Guide.*

See the figure below for a sample use of echouid:



```
G:\abcd>echouid.exe -s lservrc1

  Sentinel LM 7.3.0 Upgrade Code Generator Tool
  Copyright (C) 2003 Rainbow Technologies, Inc.

Select one of the options below to find the base license you want
to upgrade. After you have selected a base license, an upgrade code
will be generated that you can send to your software vendor so an
upgrade license can be created for you.

                 [ECHOUID SEARCH MENU]
1. Browse through licenses
2. Search by license details
3. Search by order in license file (line number)
4. Quit.
Select Option (1-4) :1
License Code: "GTIPUEBXAAAL36CR6BS75N2EZMC2YWJ5OTCN94Q7Y2EADOEHSZUGHLXK8PR5JPLD#
 "FEATURE.0" version "1.0", expires Midnight of Dec 31, 2007, additive"

 License Type                : Normal      Network         Capacity
 Feature name                : "FEATURE.0"
 Feature version             : "1.0"
 Capacity Type               : Pooled
 License Capacity            : 28600
 Max concurrent users        : 1
 Expiration date             : Midnight of Dec 31, 2007
                   [License Details Sub-Menu]
N - Next                    R - Return to search menu
P - Previous                Q - Quit
G - Generate upgrade code for this license
Select Option (N/P/G/R/Q) :G
Upgrade code : W3KBK52DDVP8S9PMETK7348XDX6KKJ9WSPAJN7L9CWWME8FGJ4CWZT4UWDZ3MW75A
86SWLGJ3WWJFR3# "FEATURE.0" version "1.0", "N1", "P28600"
Would you like to save this upgrade code to a file (y/n)?y
Upgrade code saved to file "G:\abcd/ulccodes"

G:\abcd>_
```

Sentinel LM Upgrade Code Generator Tool

Tip: *Echouid.exe* can be executed by double-clicking on its file name. However, we suggest you to evoke it from the command line because if executed by double-clicking it will not allow you to specify command line options.

After you have received the upgrade lock code you need to generate the upgrade license code for your customers.

For details on generating the upgrade license refer to Chapter 8, “Creating and Upgrading License Codes,” on page 79.

Chapter 13

Utilities

There are a number of utilities that provide information about the Sentinel LM license server, the license codes, and the network. Some of these utilities are intended for use by your customers and can be shipped to them. Others are intended for your use alone.

The utilities marked in the table below as “Customer” will be useful to your customers and can be shipped to them. Detailed information on using these utilities is contained in the *Sentinel LM System Administrator’s Online Guide*. Remember to ship the *lsapiw32.dll* file with the Windows utilities. This DLL must be present for these utilities to work.

The utilities marked as “Developer” are reserved for your use, and should not be shipped to your customers.

⚠ Do *not* ship the license code generator utilities to your customers. In fact, distribute them very carefully within your own organization to protect the security of your application.

In addition to the utilities listed below, note that Sentinel LM also provides reporting capability through a Microsoft Access database table, *ReportDB.mdb,* that allows developers and their customers to view license server log file output in bar chart form (for example, to see total use of a feature during a particular time period). To use the reporting feature, the lsusage or vusage -c option must be used to create a Comma-Separated

Values file from the log file. For information on the vusage -c option, See “Display Encrypted/Readable License Server Log File (vusage)” on page 212. For information on the Sentinel LM reporting capability, see the *Sentinel LM System Administrator’s Online Guide*.

Summary of Utilities

The table below summarizes the various utilities available for developers and customers:

cgtamp	
Description	Creates a temporary license code to ship with timefix.
Used By	Developer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	“Integrated Cleaning Tool (lsclean and lscgcln)” on page 202.
Windows Location	\tools
echoid	
Description	Obtains the information on the customer site to be used as input to the license code generator.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	See “Determine a Computer Locking Code (echoid)” on page 204. Also see the *Sentinel LM System Administrator’s Online Guide*.
Windows Location	\tools
ipxecho	
Description	Displays the IPX network address of the computer on which it is run. Used to determine the IPX address of a license server computer.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX

Where Documented	See “Display the IPX Network Address (ipxecho)” on page 205. Also see the *Sentinel LM System Administrator’s Online Guide*.
Windows Location	*\delivery\admin.net\win32*
lcommute	
Description	Checks a commuter license authorization in or out.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	See the *Sentinel LM System Administrator’s Online Guide*.
Windows Location	*\delivery\admin.net\win32*
loadls	
Description	Used with Windows NT/2000/XP/2003 only. Install and start the service.
Used By	Customer
Platforms	Windows NT/2000/XP/2003
Where Documented	See “Install the License Server as an NT/2000 Service (loadls)” on page 205. Also see the *Sentinel LM System Administrator’s Online Guide*.
Windows Location	*\delivery\server\winnt*
lscgen	
Description	Generates the license code. Command-line version of **WlscGen**.
Used By	Developer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003
Where Documented	See Chapter 8, “Creating and Upgrading License Codes,” on page 79.
Windows Location	*\tools*
lsdecode	
Description	Decrypts part of the information in a license code.

Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	See “Decrypts and Displays License Codes (lsdecode)” on page 206.
Windows Location	*\tools*
lserv, lserv9x, lservnt	
Description	Starts the license server on UNIX, Windows 95/98/ME or Windows NT/2000/XP/2003, respectively.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	See “Starting the License Server (lserv, lserv9x, lservnt)” on page 206. Also see the *Sentinel LM System Administrator’s Online Guide*.
Windows Location	*\delivery\server\winnt* and *win9x*
lslic	
Description	Installs the customer license. Accepts the license code as input or a file containing codes.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	See “Display the IPX Network Address (ipxecho)” on page 205. Also see the *Sentinel LM System Administrator’s Online Guide*.
Windows Location	*\delivery\admin.net*
lsmail	
Description	Enable user e-mail alerts for Windows users. Notice that users do not call this utility directly other than to setup the mail server address.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003

Where Documented	See "Send e-mail alert using SMTP (lsmail)" on page 209. Also see the *Sentinel LM System Administrator's Online Guide.*
Windows Location	*\delivery\server\winnt* and *win9x*
lsmon	
Description	Monitors license server transactions. Command-line version of WlmAdmin.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	See "Monitor License Server Transactions (lsmon)" on page 210. Also see the *Sentinel LM System Administrator's Online Guide.*
Windows Location	*\tools*
lspool	
Description	Monitors and dynamically configures a redundant license server pool.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	See "Maintaining the Redundant License Server Pool (lspool)" on page 210. Also see the *Sentinel LM System Administrator's Online Guide.*
Windows Location	*\tools*
lsrvdown	
Description	Shuts down license server.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	See "Shuts Down the License Server (lsrvdown)" on page 211. Also see the *Sentinel LM System Administrator's Online Guide*.
Windows Location	*\delivery\admin.net\win32*

lsusage	
Description	Displays information from a license server logfile.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	See “Display Usage Log File (lsusage)” on page 211. Also see the *Sentinel LM System Administrator’s Online Guide.*
Windows Location	\delivery\server\winnt
lsver	
Description	Displays version of Sentinel LM that is running.
Used By	Developer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	See “Display Sentinel LM Version Number (lsver)” on page 211.
Windows Location	\tools
lswhere	
Description	Displays the names and/or IP/IPX data of the computers on the network running the license server.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	See “Display List of License Servers (lswhere)” on page 211. Also see the *Sentinel LM System Administrator’s Online Guide*.
Windows Location	\delivery\admin.net\win32

rlftool	
Description	Creates and edits the redundant license server configuration file.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	See the *Sentinel LM System Administrator’s Online Guide.*
Windows Location	*\tools*
vusage	
Description	Displays a report of information from an encrypted and readable license server log file.
Used By	Developer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003, UNIX
Where Documented	Discussed later in this chapter. See “Display Encrypted/ Readable License Server Log File (vusage)” on page 212.
Windows Location	*\tools*
Wcommute	
Description	Windows tool for checking a commuter license authorization in or out.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003
Where Documented	See “Wcommute” on page 218. Also see the *Sentinel LM System Administrator’s Online Guide.*
Windows Location	*\delivery\admin.net\win32*

Wechoid	
Description	Debugging tool that displays fingerprint criteria values.
Used By	Developer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003
Where Documented	Discussed later in this chapter. See “Display Fingerprint Criteria (Wechoid)” on page 219.
Windows Location	\tools
WlmAdmin	
Description	Windows license server status and administrator program.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003
Where Documented	See “Administer License Management Activities (WlmAdmin)” on page 221. Also see the *Sentinel LM System Administrator’s Online Guide*.
Windows Location	\tools
WlscGen	
Description	Windows license code generator and upgrade license code generator
Used By	Developer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003
Where Documented	See Chapter 8, “Creating and Upgrading License Codes,” on page 79.
Windows Location	\tools
WlsGrMgr	
Description	Windows reservation file editor.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003

Where Documented	See “Editing the Reservation File on Windows (WlsGrMgr)” on page 222. Also see the *Sentinel LM System Administrator’s Online Guide*.
Windows Location	*\tools*
Wrlftool	
Description	Windows redundant license file configuration utility.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003
Where Documented	See the *Sentinel LM System Administrator’s Online Guide*.
Windows Location	*\tools*
echouid	
Description	Generates upgrade lock code of a license, to be used as input to the upgrade license code generator for creating an upgrade license.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003
Where Documented	See the *Sentinel LM System Administrator’s Online Guide*.
Windows Location	*\tools*
ulscgen	
Description	Upgrade the version or capacity or both(version and capacity) of your base license.
Used By	Developer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003
Where Documented	See the “The Upgrade License Code Generator Command Line Version - Ulscgen” on page 129.
Windows Location	*\tools*

ulsdcod	
Description	Decrypts part of the information in upgrade license code.
Used By	Customer
Platforms	Windows 95/98/ME, Windows NT/2000/XP/2003
Where Documented	"Displaying the Upgrade License Code" on page 132
Windows Location	\tools

Integrated Cleaning Tool (lsclean and lscgcln)

Sentinel LM guards against a customer trying to re-use an expired license. Consider the case where a customer receives a license that will expire on a specific calendar date. The customer may set the computer's system clock back to an earlier date to try to use the expired license (called *time tampering*). Sentinel LM detects this and prevents it.

There are situations in which the developer may wish to "clean" the customer's computer to allow the customer to re-use a license.

Note: Previous versions of Sentinel LM provided multiple tools to handle clearing the customer's computer (prsclean and timefix). These tools have been combined into a single utility, lsclean.

lsclean clears the computer of:

- Trial license use information
- Time tampering information
- Commuter license use information

The lscgcln utility command-line takes the following form:

```
lscgcln option output-file-name
```

where *option* may be:

Option	Description
-p	Creates a license directing lsclean to clear the computer of trial license expiration information.
-c	Creates a license directing lsclean to clear the computer of commuter license expiration information.
-t	Creates a license directing lsclean to clear the computer of time tampering detection information.

and *output-file-name* gives the name of the new or existing license file that will hold the license.

To create the license:

1. Open an MS-DOS window on a Windows computer.

2. Type **lscgcln** followed by the appropriate option and output file name. For example:

   ```
   lscgcln -t timecln
   ```

   This command tells lscgcln to create an lsclean license that will allow lsclean to clear time tampering information from the computer and to place the license in a file named *timecln*.

3. lscgcln will prompt you for the starting date, the number of times the license can be used, and locking information so that the license will be locked to a specific computer.

Using lsclean

To use the lsclean utility on the customer's computer:

1. Copy the lsclean utility and the license file containing the license created by lscgcln to a directory on the customer's computer.

2. At a command line prompt, type **lsclean** followed by the name of the license file containing the license created by lscgcln:

```
lsclean timecln
```

A log file will be created named *clean.log* that contains the details of the success/failure of lsclean to clear the computer of the specified information.

Tip: When clearing information used by network licenses, you must stop and restart the license server after the computer is cleaned for the change to take effect.

Note: lsclean runs on both Windows and UNIX computers.

Determine a Computer Locking Code (echoid)

The echoid command determines a computer's locking code based on the computer fingerprint. The computer fingerprint determines one computer from another and allows you to lock an application or license server to a specific computer.

The locking code echoid computes is based on the fingerprinting criteria you select by modifying the *echoid.dat* file. (For information on setting fingerprinting criteria, see Chapter 7, "Planning Customer Computer Fingerprinting," on page 73. That section also tells you how to modify the *echoid.dat* file.)

Note: The *echoid.dat* file must be in the same directory as the echoid utility.

To use echoid on UNIX, Windows 95/98 and Windows NT 4.0/Windows 2000/Windows XP and Windows 2003, at the command line prompt enter:

```
echoid
```

echoid displays the output on console.

This is the value you specify in the license code generator if you want to lock to this particular computer. If the *echoid.dat* file specifies two fingerprints, you see two locking codes listed. (For details on the use of two fingerprints, see Chapter 7, “Selecting a Primary and Secondary Fingerprint,” on page 75.)

Display the IPX Network Address (ipxecho)

The ipxecho utility 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. 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
```

If IPX protocol is used for client-server communication then the LSPROTOCOL environment variable must be set to IPX on the client end.

For details on ipxecho utility refer to the *Sentinel LM System Administrator’s Online Guide*.

Check Commuter License Authorization (lcommute)

lcommute is used to check out a license authorization from license servers. It can be used on the Windows or UNIX operating system and is functionally similar to Wcommute.

For more information on lcommute refer to the *Sentinel LM System Administrator’s Online Guide*.

Install the License Server as an NT/2000 Service (loadls)

loadls installs and uninstalls the Sentinel LM license server into the system services registery. loadls is run by a user on a license server computer. The user must have administrative privileges to load or remove an NT/2000/XP/2003 System Service.

For details on loadls utility refer to the *Sentinel LM System Administrator's Online Guide*.

Generate License Code (lscgen)

lscgen is the command-line version of the license code generator can be used at the operating system command prompt.

lscgen checks if a required input is available from the entered command line. If the input is available, it ignores the corresponding entry in the global defaults file. However, if the input is not available on the command-line, it checks the global defaults file.

For details on lscgen refer to "The License Code Generator Command-line Version – lscgen" on page 112 and also see "Interactive Input" on page 122.

Decrypts and Displays License Codes (lsdecode)

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

The lsdecode utility runs from the operating system command prompt by an administrative or application user on a stand-alone, client, or license server computer.

You can choose NOT to ship this to the customer if you do not want the customer to see the license code.

Starting the License Server (lserv, lserv9x, lservnt)

The lserv, lserv9x and lservnt utilities start 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, the LSERVOPTS environment variable should be used.

lserv, lserv9x, lservnt utilities run from the operating system command prompt by an administrative user on license server computers.

For details on lserv, lserv9x, lservnt utilities refer to the *Sentinel LM System Administrator's Online Guide*.

Install a License Code (lslic)

The lslic command adds or deletes one or more license codes without stopping or restarting the license server. At the operating system command prompt, type:

```
lslic [options]
```

where one of the following options must be specified:

Option	Description
-a *license-code*	Adds a license code to the contacted license server without updating the license file. (Change is temporary until license server is restarted.)
-A *license-code*	Adds a license code to the server's license file (*lservrc*) and communicates the change to the contacted license server. (Change is permanent.)
-d *feature version [capacity]*	Deletes a feature/version/capacity from the contacted license server but does not update the license file. (Change is temporary until the license server is restarted.)
-f *filename*	Adds all the license codes in the supplied file to the contacted license server without updating the license file. (Change is temporary until the license server is restarted.)
-F *filename*	Adds all license codes in the supplied file to the license server's license file (*lservrc*) and communicates the new information to the contacted license server. (Change is permanent.)

Option	Description
-ad *license-code dist-crit*	Adds a license code to the redundant license servers without updating the redundant license file. 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, lslic will prompt for that information.
-Ad *license-code dist-crit*	Adds a license code to the redundant license servers and updates the redundant license file (*lservrlf*). 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. lslic can not delete upgraded licenses
-fd *filename*	Adds all the license codes and distribution criteria in the file to the redundant license servers memory. (Change is temporary until the license server is restarted.)
-Fd *filename*	Add all license codes and distribution criteria in the file to the redundant license server and updates the redundant license file (*lservrlf*). (Change is permanent.)
-removeall	Removes all license codes from the license server but does not delete them from the license file (*lservrc* or *lservrlf*). (Temporary change.)
-removeallcap *feaure version*	Removes capacity for specific license from the license server but does not delete them from the license file.

For stand-alone applications, lslic communicates with the license management functions embedded in the application, not an external license server.

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: distribution criteria for license code 1

license code 2
#DIST_CRIT: distribution criteria for license code 2
```

And so on. If a license code is not followed by a #DIST_CRIT: line, 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
```

Send e-mail alert using SMTP (lsmail)

lsmail is automatically called by the license server to send e-mail when the alert action specified in the license server configuration file occurs. The file *lsmail.exe* should be in the directory in which the license server resides.

On running for the first time lsmail asks for the configuration information: the SMTP server address (the mail server), the port number to be used (default is 25), and the sender name (default is Sentinel LM).

For more details on loadls utility refer to the *Sentinel LM System Administrator's Online Guide.*

Monitor License Server Transactions (lsmon)

lsmon retrieves information about all features currently licensed by the Sentinel LM license server and clients using those features. If *Server-host* is omitted, lsmon attempts to talk to the Sentinel LM 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 attempts to contact the license server using the broadcast mechanism. If lsmon fails to find a Sentinel LM license server, it displays an error message and exits.

On UNIX computers, you can use *Poll-interval-in-seconds* to specify that lsmon should keep monitoring and reporting usage activity instead of displaying information once and stopping. If specified, lsmon waits for that many seconds between re-polls of the license server. lsmon 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 are displayed.

For more details on lsmon utility refer to the *Sentinel LM System Administrator's Online Guide.*

Maintaining the Redundant License Server Pool (lspool)

After the redundant license servers have been set up, lspool or WlmAdmin can be used 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.

For more information on lspool, refer to the *Sentinel LM System Administrator's Online Guide.*

Shuts Down the License Server (lsrvdown)

lsrvdown shuts down the named license server.

It is run from the operating system command prompt by an administrative user on a client or license server computer. On Windows 95/98/ME, only the 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/XP/2003, 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.

For more information on lsrvdown, refer to the *Sentinel LM System Administrator’s Online Guide*.

Display Usage Log File (lsusage)

lsusage displays a summary of application usage, providing information on license transactions contained in the license server usage file. It runs from the operating system command prompt by an administrative or application user on a stand-alone, client, or license server computer.

For more information on lsusage utility refer to the *Sentinel LM System Administrator’s Online Guide*.

Display Sentinel LM Version Number (lsver)

At the operating system command prompt, type: **lsver** to display the Sentinel LM version number.

Display List of License Servers (lswhere)

lswhere displays 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.

The lswhere utility is run from the operating system command prompt, by an administrative or application user on a stand-alone, client, or license server computer.

For more information on lsusage utility refer to the *Sentinel LM System Administrator's Online Guide*.

Create and Modify Redundant License Server File (rlftool)

rlftool is a command-utility for Windows and UNIX. It is used to create a new redundant license file, from DOS type:

```
rlftool
```

It can also be used to modify an existing redundant license file, type:

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

For more information on rlftool, refer to the *Sentinel LM System Administrator's Online Guide*.

Display Encrypted/Readable License Server Log File (vusage)

The vusage utility displays a report of the information in the specified encrypted license server log file. The license server log file contains a record of license transactions made by that license server. vusage displays the same information as the lsusage utility which is used at your customer's site to monitor license server actions, however it is able to read and display data encrypted to level 3 or 4 in a license server log file, which lsusage is not. You may also use the -d option to create a new, unencrypted log file from the encrypted log file. See “Creating an Unencrypted Log File” on page 216 for details.

Encrypting license code entries in the log file prevent them from being viewed or modified by your customer. For more information on setting encryption for license transactions recorded in the log file, see Chapter 8, "Log File Encryption," on page 99.

To use vusage, at the operating system command prompt type:

```
vusage [option]
```

where one or more of the following options must be specified:

Option	Description
-l *log-file*	Selects the license server log file to display. Must be specified. If this is the only option, the decrypted summary will be displayed on the screen.
-d *log-output-file*	Specifies the name of the output file to which the decrypted output will be written.
-c *CSV-format-file*	Specifies the name of an output file to which the decrypted output will be written in Comma-Separated Values (CSV) format. This CSV file can be used with Microsoft Access to produce graphical reports. For more information, see the *Sentinel LM System Administrator's Online Guide*.
-f *feature1[,version1]:* *feature2[,version2]*	Specifies feature(s) and version(s) to be included in the output. Note that *version* is optional. If this option is not supplied, the output will contain all features and versions in the log file.
-m *start-month*	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 *start-day*	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.

Option	Description
-y *start-year*	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 *end-month*	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 *end-day*	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 *end-year*	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.

Note: vusage decrypts transactions for license codes set to all levels of encryption, and will warn if a record has been tampered with:
On line number: 1ERROR: Line is tampered for session number 55
The log file may contain entries for other developers. **vusage** can only decrypt entries for your own licenses.

The vusage display looks something like this:

```
    Application Usage Summary

    --------------------------
```

```
%age        %age        Total     %age       %age       Min. App    Avg. App.    Max.App.    Total
denied      issued      tokens    queued     Qrequest   diuration   duration     duration    Capacity
                        Issued    granted               (min)       (min)        (min)       issued

For feature name: bounce, Version: Non-capacity
```

```
0.00        100.00      2         0.00       0.00       2.35        3.08        3.83        -

For feature name: bounce, Version: Capacity Non-pooled 4000

0.00        100.00      2         0.00       0.00       2.35        3.08        3.83        8000

Log report for sessions: 0
```

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.
Total tokens 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.

Element	Description
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.
Total Capacity Issued	The total amount of capacity issued, applicable only for capacity license.
LOG REPORT FOR Sessions: x	The session numbers for this license server that were logged in this file. The developer will want to track the session numbers when using encrypted log files for “pay per use” billing to verify that no session numbers were omitted by the end user.

Creating an Unencrypted Log File

In addition to creating a report from an encrypted log file, you can also use the vusage -d option to create an unencrypted version of that log file. (You might want to do this if you have your own utilities that create a report from a license server log file.)

A log file that contains encrypted entries might look like the following. In this case, an encryption level of 4 has been used and all license transaction data appears in encrypted form:

```
# Startup Sentinel LM v7.00 Fri Aug 13 17:33:37 1999
934590817 jsmith 141 4294489039

4 0 MTI1
RnJpIEF1ZyAxMyAxNzozMzozOSAxOTk5ICA5MzQ1OTA4MTkgOTkgd
iAwIDEgMCBtcmVkZGluZyBtcmVkGluZyA3
LjAwIDEgLSAtIC0gLSAtIC0= MA== 1795899 OTM2Mzg2NzE4

4 0 MTI1
RnJpIEF1ZyAxMyAxNzozMzozOSAxOTk5ICA5MzQ1OTA4MTkgOTkgd
iAyIDAgMCBtcmVkZGluZyBtcmVGluZyA3L
jAwIDEgLSAtIC0gLSAtIC0= MQ== 2582281 NDM3ODE4MA==

# Shutdown Sentinel LM v7.00 Fri Aug 13 17:33:40 1999
934590820 jsmith 4294489039 Mg== 3305931
```

You can use vusage to create a new log file in which the encrypted entries have been decrypted. For example, if the above log file is *lserv.log*, and you want to create an unencyrpted version of it named *lserv2.log*, type:

```
vusage lserv.log -d lserv2.log
```

Here is what the converted log file looks like, with encrypted transaction data converted to plain text:

```
# Startup Sentinel LM v7.00 Fri Aug 13 17:33:37 1999
934590817 jsmith 141 4294489039

1 1 MA== Fri Aug 13 17:33:39 1999 934590819 99 v 0 1 0
jsmith jsmith 7.00 1 - - - - - - MA== 764248
OTM1MzU1MDY3

1 1 MA== Fri Aug 13 17:33:39 1999 934590819 99 v 2 0 0
jsmith jsmith 7.00 1 - - - - - - MQ== 914504
MTY3ODc1Mg==

# Shutdown Sentinel LM v7.00 Fri Aug 13 17:33:40 1999
934590820 jsmith 4294489039 Mg== 3305931
```

For details on the layout of the license server usage log file, see the *Sentinel LM Programmer's Reference Manual*. However, here is a brief summary of the log file elements:

Elements	Description
Server-LFE	Customer-defined log file encryption level as specified by the license server -lfe startup option.
License-LFE	Developer-defined log file encryption level as specified during license code generation. 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.

Elements	Description
Trans	The transaction type. 0 indicates an issue, 1 a denial, and 2 a return.
Numkeys	The number of licenses in use after the current request/ release.
Keylife	The time, in seconds, that the license was issued.
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 Sentinel LM license server.
Currency	The number of licenses handled during the transaction.
Comment	Log file comment.

If the Sentinel LM 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 or vusage -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 the *Sentinel LM System Administrator's Online Guide*.)

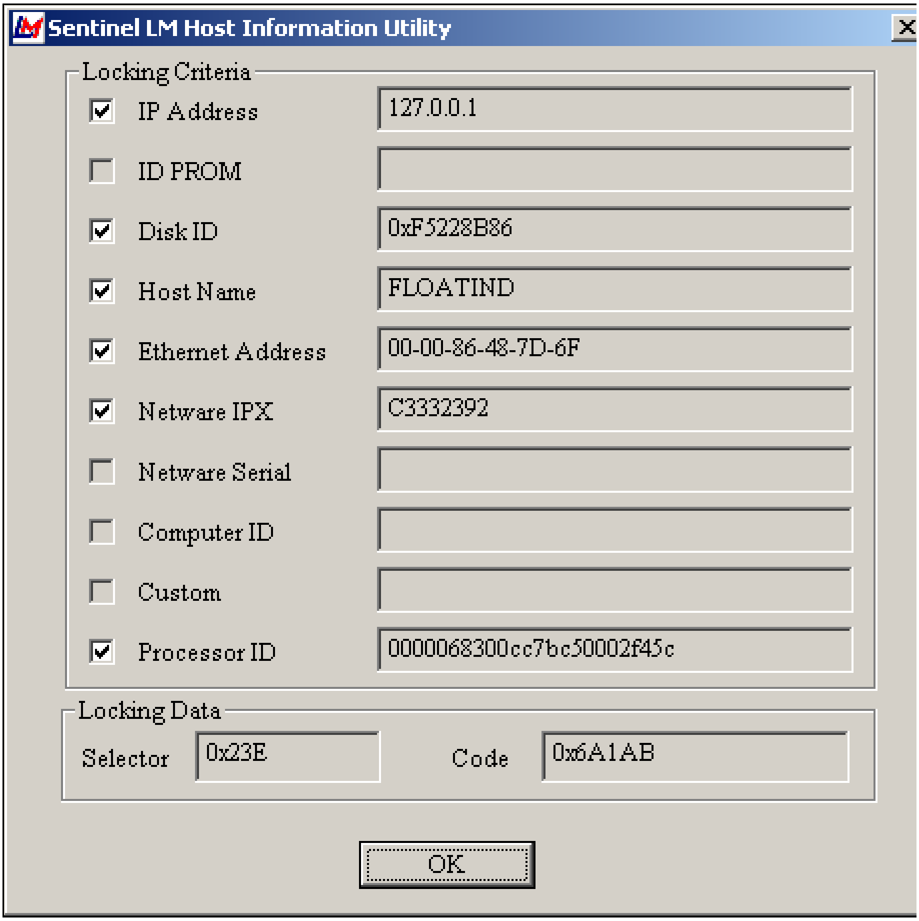
Wcommute

Wcommute (for Windows) or lcommute (Windows or UNIX) can be used 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.

For more information on Wcommute refer to the *Sentinel LM System Administrator's Online Guide*.

Display Fingerprint Criteria (Wechoid)

The Wechoid command is a Windows utility to help you debug fingerprinting. It lists all selectable fingerprint criteria. (Only those criteria applicable to the computer on which you are running the command are enabled). By selecting the check boxes for the criteria you want to use, you can see what the hexadecimal value is for those criteria. Use this value in the *echoid.dat* file you ship to your customer. To use Wechoid, click **Start** and point to *Rainbow Technologies> Sentinel LM 7.x.x > English> Show Machine ID*.



Locking Criteria

☑ IP Address: 127.0.0.1

□ ID PROM:

☑ Disk ID: 0xF5228B86

☑ Host Name: FLOATIND

☑ Ethernet Address: 00-00-86-48-7D-6F

☑ Netware IPX: C3332392

□ Netware Serial:

□ Computer ID:

□ Custom:

☑ Processor ID: 0000068300cc7bc50002f45c

Locking Data

Selector: 0x23E

Code: 0x6A1AB

Sentinel LM Host Information Utility

In the example above, IP address, Disk ID, Host name, and Ethernet address are all selected as fingerprinting criteria. The value to place in the *echoid.dat* file to select those criteria is shown in **Selector**. A fingerprint computed using these criteria generates the locking code shown in **Code**. For details on the *echoid.dat* file, see Chapter 7, “How to Select Fingerprint Criteria,” on page 74.

Tip: The machine elements are unavailable if they do not exist in this computer.

Administer License Management Activities (WlmAdmin)

WlmAdmin provides information on licensing activities. It provides information on license servers detected, details on active licenses, and information on licensed users.

WlmAdmin also performs a variety of administrative functions like:

- Monitoring Licenses and License Servers
- Editing the Reservation File
- Editing the Redundant License File
- Shutting Down the License Server

WlmAdmin utility is 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.)

For more details on WlmAdmin refer to the *Sentinel LM System Administrator's Online Guide*.

WlscGen

WlscGen is the Windows-interface version of the license code generator and upgrade license code generator.

For a comphrensive study of WlscGen refer to "Using the License Code Generator – WlscGen" on page 90.

Editing the Reservation File on Windows (WlsGrMgr)

On Windows computers, the WlsGrMgr utility can be used to create or edit a reservation file. For UNIX computers, you will need to edit the reservation file manually rather than using the WlsGrMgr utility.

For more details on WlsGrMgr refer to the *Sentinel LM System Administrator's Online Guide*.

Create and Maintain Redundant License File (Wrlftool)

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

For more details on Wrlftool refer to the *Sentinel LM System Administrator's Online Guide*.

Determine a License Upgrade Lock Code (echouid)

The echouid command generates the license upgrade lock code, which is to be used to create a upgrade license code.

For more details on echouid refer to "Generating an Upgrade Lock Code" on page 189.

Upgrade License Code Generator Command Line Version (ulscgen)

Upgrade license code can be generated with the command line version that is ulscgen.

For more details on ulscgen refer to "The Upgrade License Code Generator Command Line Version - Ulscgen" on page 129.

Decrypts and Displays Upgrade License Codes (ulsdcod)

Ulsdcod utility decrypts part of the information in upgrade license code strings. This utility is useful in determining the details of licensing agreements. It also enables you to decipher unknown codes.

You can run ulsdcod utility runs from the operating system command prompt by an administrative or application user on a stand-alone, client, or license server computer.

For more details on ulsdcod see “Displaying the Upgrade License Code” on page 132.

Tip: You can choose NOT to ship this to the customer if you do not want the customer to see the license code.

Chapter 14

Product Add-ons

Several Rainbow products work with Sentinel LM to increase your productivity and make life easier for your development team and your customers. The sections below give you an idea of the type of additional Sentinel LM products that are available. For more information, contact your Rainbow representative.

Computer ID Keys

Computer ID keys provide the best computer fingerprint available:

- **Pre-activation of Licenses** removes the license activation step and overhead.
- **Portable**, fault-tolerant fingerprint can provide backup or be moved between computers.

Sentinel Express

Sentinel Express provides the best form of license activation using the Internet:

- **End-user Registration** information captured in any of several SQL databases.

- **Cheapest Activation** method available.
- **Demo and Online Payment** options for product delivery.

Sentinel Express provides everything you need to securely offer your Sentinel LM-protected applications through the World Wide Web. Your customer can perform the following operations through your Web site:

- Select a product.
- Register with your corporate database.
- Download and activate demo software.
- Purchase the product.
- Activate the protected application.

Existing Hardware Keys

Existing Hardware keys used as computer fingerprints:

- **Use Investment** in existing hardware keys from any vendor.
- **Improve Fingerprint** by locking licenses to an external, developer controlled hardware device.

Developer Assistance from Rainbow

Developer Assistance through Rainbow's Professional Services Group:

- **Experienced** developers to assist in license strategy, advanced security and implementation.
- **Fast and Reliable** solutions from the #1 Security provider allow you to concentrate on your business.

Appendix A

Readable License Codes

This appendix contains information on expanded and concise readable license codes.

License codes can either be encrypted or readable. A readable license code stores information about the provisions of a licensing agreement in readable form, such as plain text with white spaces so that it is easily read (and understood) by the user. A checksum generated on this text serves to verify that the original string is not modified by malicious users.

The code generator generates two types of readable license codes:

- Concise readable license string
- Expanded readable license string

If a license option is not chosen when generating the license code, “NiL” is written in the readable string at the corresponding position for Expanded strings.

Concise Readable String

In this format of readable codes, the license string is small in length (approximately 100 characters). Further, some numeric values (e.g., lifetime, holdtime, units etc.) do not have the appropriate word appended to them. This is a variable format string, where irrelevant data is not written out at all.

Note: Concise readable string are not available for capacity licenses.

Expanded Readable String

This is a fixed format readable license string, where the reserved value "NiL" is present for fields that are not allowed to be entered while running the code generator due to the specific options that have been chosen. For numeric values this format has appropriate words (e.g. -Minutes, -Units, -Keys etc.) appended to them to make the string more meaningful to the reader. This is a fixed format string which is easy to parse by other tools, since the value contained at position *n* is always known beforehand.

To provide you with greater flexibility, utilities have been developed to convert readable code to encrypted code (rdctoenc) and vice-versa (enctordc). These should *not* be shipped to the end users. They are only for your use.

Note: Expanded readable string are not available for capacity licenses.

For readable codes, the utilities lsdecode and lslic behave in the same manner as for encrypted codes.

Concise vs. Expanded License Strings

There are some differences between a concise readable string and an expanded readable license string.

For example, for the following input:

```
Feature_name = DOTS
code_type = SHORT
Additive/exclusive: additive
No restriction on the number of concurrent keys.
No expiration date.
sharing criteria = no sharing.
client_server_lock_mode = demonstration mode.
key lifetime = 60 minutes
```

The expanded readable license string would be:

```
2 DOTS 0 SHORT ADD 65535_KEYS NEVER NO_SHR DEMO NiL
NiL NiL NiL NiL 60_MINS NiL NiL JDOWJ7VBNP
```

The concise readable license string would be:

```
2 DOTS 0 SHORT ADD 65535 NEVER NO_SHR DEMO 60 YZNRJ7S2
```

NiLs appear in the expanded license string for all those values that are not allowed for that type of a licensing agreement. No NiLs appear in the concise format.

In the expanded license string, a string is appended to the numeric values to specify what that numeric value stands for, e.g., 60_MINS implies that 60 specifies the time in minutes. These strings do not appear in the concise format, only a 60 appears in the concise readable license string in place of 60_MINS in the expanded format.

Readable Code Format

The sequence of fields in the generated readable license string is:

```
license version
feature name
feature version
short/long flag
trial/normal
standalone/network flag
additive/exclusive flag
num keys (maximum number of keys in case user does not
specify)
soft limit
lic start information
lic expiration day
lic expiration month
lic expiration year
(If the user does not specify the expiration year etc.
"NEVER" is written.)
sharing criterion:share_limit
encryption information
issue commuter licenses
```

```
redundant licenses
client_server_lock mode
server_lock info
node1_lock_info:node1_keys,node2_lock_info:node2_keys,...
site_license_subnet1,site_license_subnet2,...
holding criterion
key_holdtime
key_lifetime
vendor_info
clock tamper flag
Secret1,Secret2,... (encrypted)
checksum (encrypted)
# (The delimiter for writing comments after the valid license
string.)
comments, if any.
```

For keys per client host, the format is:

```
{hostid|IPAddress}:num_keys.
```

For example:

```
0xfa789b:4,0xa54ed892,0x1234567:30
```

The *[:n]* subfield is optional. If it is not present, it means that the number of licenses per node is unlimited.

For the sharing limit, the format is:

```
SHARE_CODE:20
SHARE_CODE is one of
{USER_SHR | HOST_SHR | XDISP_SHR | APP_SHR}
```

The *[:n]* subfield is optional. If it is not present, it means that the extent of sharing is unlimited.

Appendix B

Sentinel LM-Shell Error Messages

Sentinel LM-Shell error messages can be customized for your particular business or translated into another language.

If you use the Sentinel LM-Shell Windows-interface, you can change the text of these messages on the **User Messages** tab of Sentinel LM-Shell. Your messages will be output into a file using the same name as your project file with the extension of *.shm*.

If you use Sentinel LM-Shell in command-line mode, you can edit the *slmshell.msg* file in the *\LM-Shell* directory in the Sentinel LM installation directory. After changing the file, rename it to *lsshell.msg* or specify the file name in the /M option when invoking the command-line version of Sentinel LM-Shell.

Sentinel LM-Shell Message Display

The following error messages are displayed by Sentinel LM-Shell.

Sentinel LM-Shell Error Messages

Shell Err. N0.	Return Code	Default Message	Description
001	INIT_ERROR	Could not launch the Client Activator	Initialization Error. A problem occurred when initializing the license manager library.
002	NO_LICENSE	Could not get a license.	No license was found.
003	ERROR_LOADING	Error loading program.	The protected application was not loaded into memory prior to execution.
004	MEMORY_ACCESS _ERR	Memory access error.	Error accessing memory at run-time.
005	IMPORT_LIB_ ERROR	Error loading import library.	Memory access error. Error loading import library.
006	IMPORT_FUNC_ ERROR	Error importing library function.	Error importing library function.
007	HEARTBEAT_FAIL	Heartbeat failed.	Failed to receive a response to a periodic query from the license server.
008	LIC_RENEWAL_ FAIL	License renewal failed.	Failed to renew license or portable hardware key removed.
009	ACTIVATOR_ ERROR	Could not launch the Client Activator.	Client Activator error. Unknown error occurred when launching the Client Activator.
010	METER_UNIT	License meter driver initialization error.	License meter hardware key driver initialization error.
011	METER_VERSION	License meter version mismatch.	Meter key access error. Error accessing the hardware key which meters Sentinel LM-Shell.
012	METER_ACCESS	Meter key access error.	Unknown error occurred when launching the Client Activator.

Sentinel LM - Specific Messages

Sentinel LM Messages

Condition	Description
ACTIVATOR_ERROR	Unknown error occurred when launching the Client Activator.
ACTIVE_HANDLE	The specified handle is an active handle.
ADD_LIC_FAILURE	Failed to add license string to the server.
ALL_KEYS_RELEASED	All the keys issued to the feature have been returned.
ALL_LICS_IN_USE	All license keys are currently in use.
AMBIGUOUS_HANDLE	The status of the handle is ambiguous.
BAD_HANDLE	Bad feature handle was returned from an API call.
BAD_INDEX	Bad index returned from an API call.
CANT_RETURN_KEY	Can't return key for feature.
CLIENT_MISMATCH	Could not understand message received from the license server.
CLIENT_NOT_AUTH	Client is not authorized for the specified action.
CLIENT_NOT_FOUND	Could not find the specified client for the feature.
CLOCK_TAMPERING	Request denied due to clock tamper detection.
CNT_PROV_ERR	Unknown error code
COMM_PROBLEMS	Unable to talk to the host specified. Verify Client / Server communication.
COULD_NOT_LOCK	Feature not licensed to run on this machine.
DEL_LIC_FAILURE	Failed to delete feature from the server
ERROR_LOADING	The protected application was not loaded into memory prior to execution.
FCTN_CALL_ERROR	Error in calling the function. Check the calling parameters.

Sentinel LM Messages (Continued)

Condition	Description
FEATURE_NOT_UPDT	The updates for the specified feature have not been made so far.
FNGRPNT_MISMATCH	Machine's fingerprint mismatched.
HEARTBEAT_FAIL	Failed to receive a response to a periodic query from the license server.
IMPORT_FUNC_ERROR	Error importing library function.
IMPORT_LIB_ERROR	Error loading import library.
INIT_ERROR	A problem occurred when initializing the license manager library.
INTERNAL_ERROR	Internal error in licensing or accessing feature.
INVALID_DOMAIN	Cannot perform this operation on the domain name specified.
INVALID_TAG_TYPE	Tag type is incompatible with requested operation.
INVOKE_ACTIVATOR	Text displayed to end users when auto activation has been enabled in the protected application and a license is not found.
LIC_MULT_VENDORS	Feature licensed by multiple vendors.
LIC_OTHER_VENDOR	Feature licensed by a different vendor.
LIC_QUEUING_MSG	Text displayed to end users when license queuing is in progress.
LIC_RENEWAL_FAIL	Failed to renew license or portable hardware key removed.
LIC_STRING_UNAVL	No license string is available.
LICENSE_EXPIRED	Feature can not run due to time restrictions on it.
LICENSE _TIMED_OUT	License terminated because renewal time expired.
LOCAL_UPDATE	The last update was done locally.
LOG_FILE_UNCHANGE	Cannot change specified log filename on license server.

Sentinel LM Messages (Continued)

Condition	Description
LOG_NOT_FOUND	The specified log filename can not be found on license server.
LQ_CANCEL_INST	Instruction on how to cancel a queued task.
MEMORY_ACCESS_ERR	Error accessing memory at run-time.
METER_ACCESS	Error accessing the hardware key which meters Sentinel LM-Shell.
METER_INIT	License meter hardware key driver initialization error.
METER_VERSION	License meter hardware key version mismatch.
NO_ADDL_UNITS	No additional units are available.
NO_ERROR	No error is displayed.
NO_LIC_STRING	No license string is available.
NO_LICENSE	No license was found.
NO_LICENSE_KEY	Unable to obtain licensing key when the shelled application is executed.
NO_MORE_CLIENTS	No more clients to report.
NO_MORE_FEATURES	No more features to report.
NO_RESOURCES	Could not locate enough licensing resources.
NO_SRVRS_ON_NET	No servers on this subnet.
NO_SUCCESS	No success in achieving the target.
NON_EXISTENT_KEY	Attempt to return a non-existent key for feature.
NULL_FEATURE	Feature name or version cannot be NULL.
OUT_OF_MEMORY	No resources available.
OUT_OF_QUE_RES	Could not locate enough resources to queue for license feature.
QUEUED_HANDLE	The specified handle is a queued handle.
REMOTE_UPDATE	The last update was done remotely.

Sentinel LM Messages (Continued)

Condition	Description
SERVER_DOWN	Cannot talk to the license server. Verify server is running.
SEVERE_INTRNL_ERR	Severe internal error in licensing or accessing data.
SHUTDOWN_WARNING	Warning displayed that the application will be shut down because the license is no longer valid (e.g. expired).
SRVR_NOT_RESP	License server not responding.
STRING_TRUNCATED	Input buffer too small, string truncated.
SYS_UNAVAILABLE	Licensing system not available.
TAG_NOT_SUPPORTED	License server on host does not support tags.
TRIAL_EXHAUSTED	Duration or usage of a trial license is exhausted.
UNAUTH_OPERATION	Unauthorized operation requested.
UNK_ERR_CODE	Unknown error code.
UNK_SHARE_ID	Unknown shared id specified.
UNKNOWN_TAG	Supplied tag is not known to the license server on the host.
UNKNOWN_TAG_TYPE	Tag type is not known to server.
UNKNOWN_SERVER	Unknown server host.
UNSPECIFIED_SERVER	Server hostname not specified. Set environment variable LSHOST to name the server.
UPD_TAG_KEY_ERR	Invalid attempt to update a tagged key.
USER_EXCLUDED	User/machine excluded from running the give feature.

Index